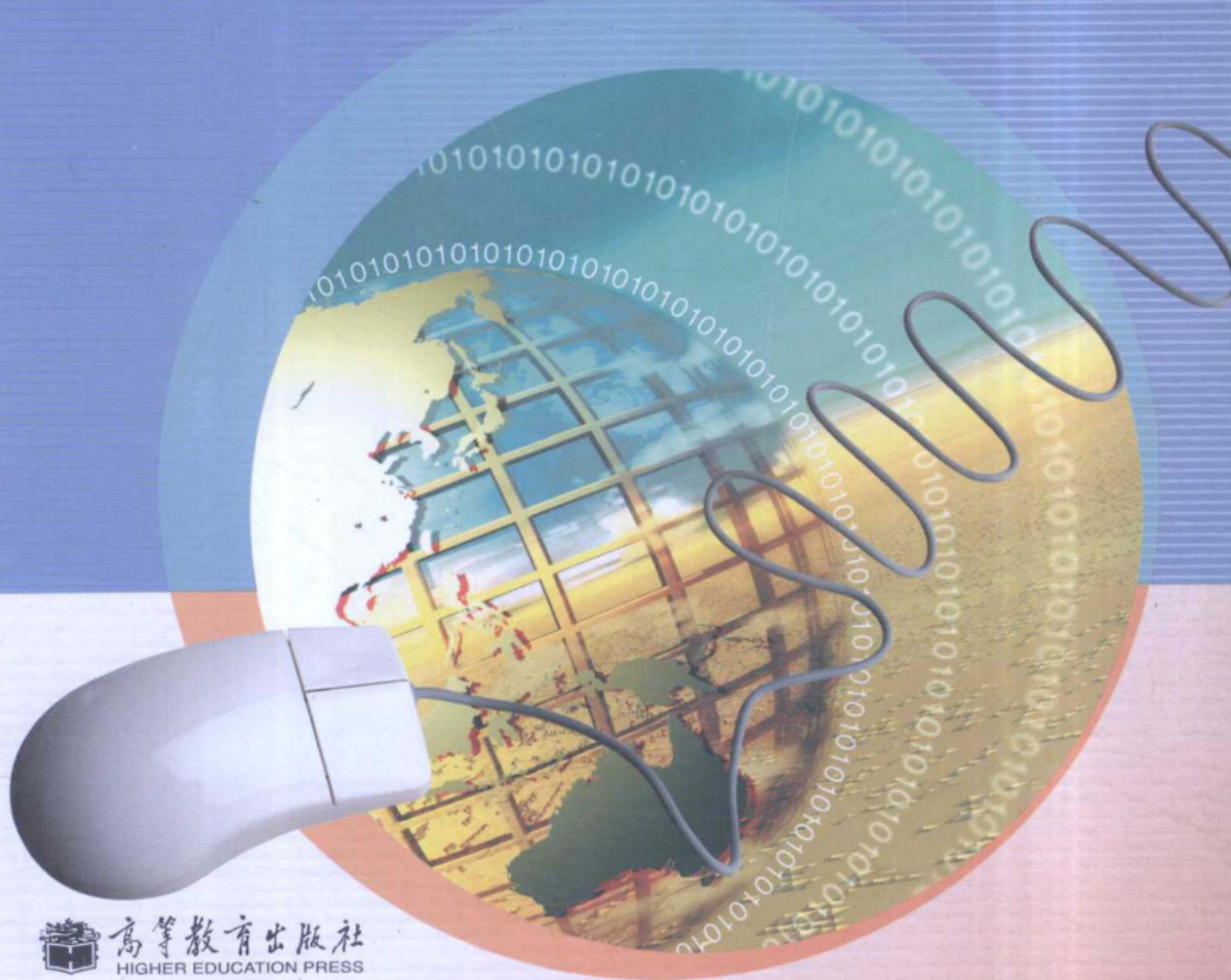
中等职业教育国家规划教材
全国中等职业教育教材审定委员会审定

可视化编程应用
—Visual Basic
（第3版）

主编 贾长云 朱香卫

（计算机应用专业）



高等教育出版社
HIGHER EDUCATION PRESS

高等教育出版社

中等职业教育计算机应用专业国家规划教材及配套教学用书目录

计算机原理（第3版）
计算机原理学习指导（第3版）（附光盘）
计算机组成与工作原理
编程语言基础——QBASIC语言（第3版）
编程语言基础——C语言（第3版）
C语言学习辅导与上机实习（第3版）
数据库应用基础——Visual FoxPro（第3版）
Visual FoxPro 学习辅导与上机实习（第3版）
数据库应用基础——Access（第3版）
Access学习辅导与上机实习（第2版）
计算机组装与维修（第3版）
计算机组装与维修实训（第3版）
计算机组装与维修学习指导（第3版）（附光盘）
计算机网络技术（第3版）
网络技术学习指导（第3版）（附光盘）
多媒体技术应用（第3版）
多媒体技术应用学习指导（第3版）（附光盘）
■ 可视化编程应用——Visual Basic（第3版）
Visual Basic 学习辅导与上机实习（第3版）
局域网组成实践（第3版）
Internet 应用（第3版）
计算机录入与排版（第2版）
办公自动化应用（第2版）
计算机专业英语（第2版）
实用计算机英语（附光盘）
中文Word 2003与Excel 2003应用基础（第2版）
常用工具软件（第2版）
电子商务概论（第2版）
办公软件应用——Office 2007（第3版）
计算机动画设计——Flash 8（第2版）
中文3ds Max 9案例应用教程
计算机平面美术设计
——Photoshop CS2和CorelDRAW 12（第2版）
计算机平面美术设计案例
——Photoshop CS2和CorelDRAW 12（第2版）
网页设计与制作——FrontPage 2003（第2版）

赠送：
100小时网上学习　免费下载多种资源
增值学习卡
网址：sve.hep.com.cn
网上学习/资源下载/权威专家在线答疑
防伪标打盗电话：10669588128O
免费查询/甄别盗版/赢取大奖
使用说明详见书内"郑重声明"页
明码 0134 3569 2743 9101
密码

ISBN 978-7-04-028684-7
9 787040 286847 >
定价 24.90元

中等职业教育国家规划教材
全国中等职业教育教材审定委员会审定

可视化编程应用
——Visual Basic

Keshihua Biancheng Yingyong——Visual Basic

（第3版）

（计算机应用专业）

主　　编　贾长云　朱香卫
责任主审　宋方敏
审　　稿　张　宏　张幸儿

高等教育出版社·北京
HIGHER EDUCATION PRESS　BEIJING

内容提要

本书是中等职业教育计算机应用专业国家规划教材，根据教育部2001年颁布的中等职业学校计算机应用专业可视化编程应用课程教学基本要求编写，同时参考了相关省市中职对口升学考试大纲。

本书针对职业教育的特点，突出基础性、先进性、实用性和操作性，注重对学生创新能力、实践能力、自学能力等各种应用能力的培养。本书以Visual Basic 6.0为平台，以初学可视化编程应用的学生为对象，介绍了VB的基础知识和基本操作方法。主要内容包括：可视化编程与VB开发环境、VB简单应用程序的设计、VB程序设计语言基础、VB中的过程、VB中的窗体与常用控件、键盘与鼠标的控制、菜单与工具栏、文件管理、图形处理与多媒体应用、数据库应用程序的开发和实验。各章配有习题，附录中提供了12个实验。

本书配套学习卡网络教学资源，使用本书封底所附的学习卡，可登录网站http://sve.hep.com.cn，获得相关教学资源。

本书适合中等职业学校计算机应用专业以及其他相关专业使用，也可作为各类计算机培训的教学用书及计算机等级考试的辅导用书。

图书在版编目（CIP）数据

可视化编程应用：Visual Basic / 贾长云，朱香卫主编. — 3版. — 北京 : 高等教育出版社，2012.2
计算机应用专业
ISBN 978-7-04-028684-7

Ⅰ. ①可… Ⅱ. ①贾… ②朱… Ⅲ. ①BASIC语言－程序设计－中等专业学校－教材 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2011)第272804号

策划编辑	郭福生	责任编辑	郭福生	封面设计	王 洋	版式设计	杜微言
责任校对	胡美萍	责任印制	田 甜				

出版发行	高等教育出版社	网 址	http://www.hep.edu.cn
社 址	北京市西城区德外大街4号		http://www.hep.com.cn
邮政编码	100120	网上订购	http://www.landraco.com
印 刷	北京嘉实印刷有限公司		http://www.landraco.com.cn
开 本	787mm×1092mm 1/16		
印 张	15	版 次	2002年8月第1版
字 数	370千字		2012年2月第3版
购书热线	010-58581118	印 次	2012年2月第1次印刷
咨询电话	400-810-0598	定 价	24.90元

本书如有缺页、倒页、脱页等质量问题，请到所购图书销售部门联系调换。
版权所有 侵权必究
物 料 号 28684-00

第3版前言

2002年本书作为中等职业教育计算机应用专业国家规划教材正式出版，受到了职业院校广大师生的欢迎。2006年在听取广大读者建议与意见的基础上，本书进行了第一次修订。经过5年的使用，我们组织相关人员对本书进行了再次修订。

本次修订的基本原则是以职业教育计算机应用专业的基本教学要求为基准，围绕中等职业学校对口升学考试对计算机语言的要求，进一步降低教学内容的难度，以够用、实用为原则。修订主要体现在以下几个方面：

1. 对教材中存在的文字及表述错误进行了仔细的修正，使其更适合中职学生阅读与学习。

2. 教材中内容较难超出教学基本要求的部分采用加“*”的方法进行了标识，供各学校在教学时根据实际情况选讲，如控件数组、通用对话框、图形处理、拖放操作等。

3. 考虑到数据库方面的知识在其他课程也要涉及，第10章标记为选讲内容。

4. 对附录中实验的内容也进行了相应的调整与修改。

本书由贾长云和朱香卫任主编，贾长云负责总体安排并修订了第1～3章，张建修订了第4章和第5章，朱锋修订了第7章和第8章，朱香卫修订了其余各章及附录。

总之，通过这次修订主要是想做到内容上与中职对口升学考试的基本要求完全接轨，体系上突出案例教学的教学要求，难度上与中等职业学生的实际水平相适应。与本书相配套的《Visual Basic学习辅导与上机实习》一书也同时进行了修订，欢迎广大读者选用。

全书建议总学时为64～72学时，实验不少于28学时。各章的学时安排推荐如下：

章　序	章　名	理论学时	实验学时
第1章	可视化编程与VB开发环境	2	2
第2章	VB简单应用程序的设计	2～3	2～4
第3章	VB程序设计语言基础	6～7	4
第4章	VB中的过程	4～5	4
第5章	VB中的窗体与常用控件	6～7	4～6
第6章	键盘与鼠标的控制	3	2
第7章	菜单与工具栏	2	2
第8章	文件管理	4	4
第9章	图形处理与多媒体应用	3	2
*第10章	数据库应用程序的开发	4	2
合计		36～40	28～32

本书配套网络教学资源，通过封底所附学习卡，可登录网站（http://sve.hep.com.cn），获取

相关教学资源。学习卡兼有防伪功能，可查询图书真伪，详细说明见书末“郑重声明”页。

由于编者水平所限，错误与疏漏之处在所难免，欢迎广大读者不吝批评指正。编者的联系方式：jcy819@gmail.com。

本书由南京大学计算机科学与技术系副主任宋方敏教授担任责任主审，由南京大学张宏教授、张幸儿教授审稿，他们为本书付出了大量辛勤的劳动，提出许多宝贵意见，使本书增色不少，在此表示感谢。

编　者

2011年9月

第1版前言

本书是中等职业教育计算机及应用专业国家规划教材，根据教育部2001年颁布的中等职业学校计算机及应用专业可视化编程应用课程教学基本要求编写，同时参考了全国计算机等级考试Visual Basic语言程序设计考试大纲。

可视化(Visual)程序设计是一种全新的程序设计方法。它一改过去效率低下的面向过程的程序设计方法，而采用面向对象的程序设计方法。现在能够提供进行可视化程序设计的开发工具有很多，如美国Microsoft公司推出的Visual Studio软件系列，Borland公司的C++、Delphi系列等。本书将向读者介绍如何在中文Visual Basic 6.0中进行可视化程序设计的一些基本方法。

Visual Basic是Microsoft公司推出的Visual Studio系列开发软件中的一种。它简单易学、功能强大、适应面宽，受到了广大计算机用户的欢迎，得到了广泛的应用。本书介绍了Visual Basic的基本概念和基本用法，针对Visual Basic主要应用领域，着重介绍了文件操作、数据库应用技术、ActiveX控件、DHTML程序设计技术。内容由浅入深，注重典型性和实用性相结合。本书具有以下特色：

(1)基本性。本书重点阐述了计算机面向对象程序设计的基本概念、基本原理和基本技能，同时也能反映计算机软件开发技术的最新发展状况。

由于Visual Basic的功能十分强大，应用场合也非常广泛。要想在一本普通的中职教材中进行面面俱到的讲述是不现实的。因此本教材将以Visual Basic在管理信息系统方面的应用为主要内容，对其他方面的应用只作简单介绍。

(2)典型性。本书选择的实例和实验都与实际应用相联系，所选实例具有一定的代表性、趣味性。一方面能有利于学生掌握有关的知识点，另一方面也能调动学生的学习积极性，提高学习兴趣。在本教材中，选择学生成绩管理系统作为MIS系统的实例，并充分运用SQL语言的功能强大，易于学习理解的特点。本实例中还结合了网上Web数据查询功能，使之具有典型性。

(3)精炼性。在有限的篇幅内阐明教学目标所要求的内容。文字的表达深入浅出，采用通俗易懂的语言并结合典型实例来阐述深奥的概念，如对象、事件驱动、焦点等。内容力求达到“够用、实用”，同时考虑到学生进一步学习的需要，注意培养学生主动获取知识的能力，为更进一步的学习奠定基础。

(4)逻辑性。教材内容的组织与编排既符合知识的逻辑顺序，又符合学生的思维发展规律，同时还符合Visual Basic本身作为一个软件开发工具所具有的特点。因此本教材从可视化程序设计基本知识、基本概念入手，运用Visual Basic来完成实际应用系统设计。全书的逻辑主线是完成学生成绩管理系统设计。

(5)先进性。教材的内容能及时反映当今Visual Basic发展中最先进的成果、编程方法与编程手段。对多媒体的应用、ActiveX编程以及在Internet应用等方面的内容都作了简单的介绍。

本书在教材总体结构的编排上也体现了学生学习的规律。全书共11章，按照由易到难的原则分为三个部分，第一部分(第1章至第5章)为基础部分，主要介绍VB开发环境的应用、简

单应用程序的设计和面向对象程序设计基础；第二部分（第6章至第8章）为应用部分，主要介绍可视化程序的开发方法，内容包括窗体、控件、菜单、工具栏、状态栏等；第三部分为提高部分，内容包括数据库应用技术、MIS应用系统开发、文件操作和其他应用。

全书内容的编排以培养学生的能力为本位，以任务驱动为特色。在每章开始首先明确本章学习应达到的能力目标并提出相应的任务，具体内容将从解决任务的方法出发，涉及原理性的内容一般不介绍。为便于学习和阅读，将操作技巧与注意事项分散到讲课内容之中。在每章的最后均有内容丰富，能体现本章能力目标的习题。

由于本教材的应用部分基本都是围绕一个小型的信息管理系统来展开的，因此实验部分的内容也是为此而编排的。为了便于学生的学习与教师的教学，本书将实验内容集中编排在全书的最后，每个实验的内容基本也是围绕信息系统的开发服务的，实验做完了，一个管理信息系统也就完成了。实验的设计分层次，能扎实、活泼、有序地训练学生计算机程序设计的技能，培养学生的实践能力。另外，本教材中带有“*”部分的章节难度较大，读者可根据情况自行选用。

全书建议总学时为64学时，实验不少于28学时。各章的学时安排推荐如下：

章　序	章　名	理论学时	实验学时
第1章	可视化编程概述	1	
第2章	VB的开发环境	1	
第3章	VB简单应用程序的设计	3	2
第4章	VB程序设计语言基础	3	4
第5章	VB中的过程	4	2
第6章	VB中的窗体与常用控件	5	4
第7章	菜单与工具栏	3	4
第8章	文件管理	4	2
第9章	数据库应用程序的开发	6	6
第10章	VB的其他应用	4	2*
第11章	管理信息系统的开发	2	2*
合计		36	28

本书第1～3章由麻泓编写，第4～8章由贾长云编写，第9～11章及实验由陆卫忠编写。全书由贾长云、陆卫忠统编。

由于笔者水平所限，缺点与错误在所难免，谨诚欢迎广大读者批评指正。

本书由南京大学计算机系副主任宋方敏教授担任责任主审，由南京大学张宏教授、张幸儿教授审稿，他们为本书付出了大量辛勤的劳动，提出许多宝贵意见，使本书增色不少，在此表示感谢。

作者（lyghhit_jcy@sina.com）

2001年11月

中等职业教育国家规划教材
出版说明

为了贯彻《中共中央国务院关于深化教育改革全面推进素质教育的决定》精神，落实《面向21世纪教育振兴行动计划》中提出的职业教育课程改革和教材建设规划，根据教育部关于《中等职业教育国家规划教材申报、立项及管理意见》（教职成[2001]1号）的精神，我们组织力量对实现中等职业教育培养目标和保证基本教学规格起保障作用的德育课程、文化基础课程、专业技术基础课程和80个重点建设专业主干课程的教材进行了规划和编写，从2001年秋季开学起，国家规划教材将陆续提供给各类中等职业学校选用。

国家规划教材是根据教育部最新颁布的德育课程、文化基础课程、专业技术基础课程和80个重点建设专业主干课程的教学大纲（课程教学基本要求）编写，并经全国中等职业教育教材审定委员会审定。新教材全面贯彻素质教育思想，从社会发展对高素质劳动者和中初级专门人才需要的实际出发，注重对学生的创新精神和实践能力的培养。新教材在理论体系、组织结构和阐述方法等方面均作了一些新的尝试。新教材实行一纲多本，努力为学校选用教材提供比较和选择，满足不同学制、不同专业和不同办学条件的学校的教学需要。

希望各地、各部门积极推广和选用国家规划教材，并在使用过程中，注意总结经验，及时提出修改意见和建议，使之不断完善和提高。

教育部职业教育与成人教育司

二〇〇一年十月

目　录

第 1 章　可视化编程与 VB 开发环境 …… 1

1.1　可视化编程的概念 …… 1
1.1.1　面向对象的程序设计 …… 2
1.1.2　类、对象、属性与方法 …… 2
1.1.3　事件及事件过程 …… 3
1.2　可视化程序设计的开发平台 …… 3
1.2.1　Visual Basic 开发平台简介 …… 4
1.2.2　Visual Basic 的功能与特点 …… 4
1.3　VB 开发环境初识 …… 5
1.3.1　VB 的启动 …… 5
1.3.2　开发环境的组成 …… 6
1.4　VB 快速入门 …… 8
1.4.1　案例一：设计“欢迎进入奇妙的 VB 世界” …… 8
1.4.2　VB 的开发模式 …… 12
1.4.3　VB 的工程 …… 12
1.5　获取帮助 …… 14
1.5.1　使用联机帮助 …… 14
1.5.2　从 Internet 上获得帮助 …… 16
习题 1 …… 16

第 2 章　VB 简单应用程序的设计 …… 18

2.1　VB 中的对象 …… 18
2.1.1　对象属性及其设置 …… 18
2.1.2　对象事件与方法 …… 19
2.2　窗体与控件概述 …… 20
2.2.1　窗体及应用 …… 20
2.2.2　控件与焦点 …… 22
2.2.3　控件的分类 …… 23
2.3　窗体的布局设计 …… 24
2.3.1　控件的添加/删除 …… 24
2.3.2　调整控件的大小与位置 …… 24
2.3.3　窗体与控件的布局调整 …… 24
2.4　控件的常用属性 …… 26
2.5　基本控件的使用 …… 27
2.5.1　标签 …… 28
2.5.2　文本框 …… 29
2.5.3　命令按钮 …… 32
2.5.4　案例二的进阶设计 …… 33
习题 2 …… 35

第 3 章　VB 程序设计语言基础 …… 37

3.1　模块与代码窗口 …… 37
3.1.1　模块 …… 37
3.1.2　过程 …… 38
3.1.3　代码窗口及其使用 …… 38
3.1.4　编码规则与注释 …… 39
3.2　VB 中的数据与运算 …… 40
3.2.1　数据类型 …… 40
3.2.2　常量与变量 …… 41
3.2.3　运算符与表达式 …… 43
3.2.4　数组 …… 44
3.3　标准函数 …… 46
3.3.1　常用内部函数 …… 46
3.3.2　字符处理 …… 47
3.3.3　Shell 函数 …… 48
3.4　输入与输出 …… 49
3.4.1　数据的输入 …… 49
3.4.2　数据的输出 …… 50
3.5　程序流程控制 …… 52
3.5.1　判定结构 …… 52
3.5.2　循环结构 …… 54
习题 3 …… 59

第 4 章　VB 中的过程 …… 61

4.1　过程概述 …… 61
4.2　过程定义与调用 …… 61
4.2.1　过程的定义与建立 …… 61
4.2.2　过程的调用 …… 64

4.3 过程调用中的参数传递 …… 66
4.3.1 按地址传送 …… 66
4.3.2 按值传送 …… 67
*4.3.3 数组参数 …… 67
4.4 变量与过程的作用域 …… 69
4.4.1 作用域概述 …… 69
4.4.2 过程级变量 …… 69
4.4.3 窗体/模块级变量 …… 69
4.4.4 全局变量 …… 69
4.5 程序的调试与排错 …… 71
4.5.1 程序错误类型 …… 71
4.5.2 调试与排错 …… 72
*4.5.3 出错处理 …… 75
习题4 …… 76

第5章 VB中的窗体与常用控件 …… 78

5.1 控件应用基础 …… 78
5.1.1 常用事件 …… 78
5.1.2 Tab键顺序 …… 81
*5.1.3 控件数组 …… 82
5.2 单选按钮与复选框 …… 83
5.2.1 单选按钮与复选框的创建 …… 83
5.2.2 设置单选按钮与复选框的状态 …… 84
5.2.3 单选按钮与复选框的使用 …… 84
5.2.4 框架的使用 …… 86
5.3 列表框与组合框 …… 87
5.3.1 列表框与组合框的设置 …… 87
5.3.2 列表框与组合框的使用 …… 88
5.3.3 应用举例 …… 90
5.4 其他常用控件 …… 93
5.4.1 滚动条 …… 93
5.4.2 定时器 …… 95
5.5 窗体控制与设计 …… 96
5.5.1 窗体的运行与控制 …… 96
5.5.2 窗体的设计原则 …… 102
5.6 案例三的设计 …… 103
习题5 …… 107

第6章 键盘与鼠标的控制 …… 109

6.1 键盘的控制 …… 109
6.1.1 KeyPress事件 …… 110
6.1.2 KeyDown与KeyUp事件 …… 110
6.2 鼠标的控制 …… 112
6.2.1 常用鼠标事件 …… 112
6.2.2 鼠标按键的检测 …… 113
6.2.3 鼠标与键盘的结合 …… 115
*6.3 拖放 …… 115
6.3.1 拖放概述 …… 116
6.3.2 拖放控制 …… 117
6.4 案例四设计 …… 119
习题6 …… 120

第7章 菜单与工具栏 …… 121

7.1 多文档界面 …… 121
7.1.1 MDI窗体及其子窗体的创建 …… 122
7.1.2 MDI窗体的显示 …… 124
7.1.3 MDI窗体的卸载 …… 124
7.2 菜单的规划与设计 …… 125
7.2.1 菜单的结构 …… 125
7.2.2 菜单的规划 …… 125
7.2.3 下拉式菜单的设计 …… 126
7.3 弹出式菜单 …… 131
7.4 工具栏 …… 134
习题7 …… 137

第8章 文件管理 …… 139

8.1 文件及其访问方式 …… 140
8.2 文件的处理 …… 141
8.2.1 顺序文件的处理 …… 141
8.2.2 随机文件 …… 147
8.2.3 文件处理常用命令及函数汇总 …… 152
8.3 文件管理控件 …… 153
8.3.1 驱动器列表框 …… 154
8.3.2 目录列表框 …… 155
8.3.3 文件列表框 …… 156
8.4 通用对话框 …… 159
8.4.1 通用对话框的使用与常用属性 …… 160
8.4.2 “打开”与“另存为”对话框 …… 161
8.4.3 “字体”、“颜色”与“打印”对话框 …… 162

习题 8 …… 165

第 9 章 图形处理与多媒体应用 …… 167

9.1 图形处理 …… 167
9.1.1 坐标系 …… 167
9.1.2 颜色的使用 …… 169
9.1.3 图形控件的使用 …… 170
9.1.4 常用的绘图方法 …… 173
9.1.5 简单动画设计 …… 178
9.2 多媒体应用 …… 180
9.2.1 Animation 控件 …… 181
*9.2.2 多媒体概述 …… 182
9.2.3 多媒体控件 MMControl …… 182
9.2.4 多媒体程序举例 …… 185
习题 9 …… 189

第 10 章 数据库应用程序的开发 …… 190

10.1 数据库基础 …… 190
10.1.1 数据库的概念 …… 190
10.1.2 关系型数据库基础 …… 191
10.1.3 建立数据库 …… 191
10.1.4 建立并维护数据库表 …… 191
10.2 用数据窗体向导设计数据库应用程序 …… 195
10.3 用 Data 控件访问数据库 …… 197
10.3.1 Data 控件 …… 198
10.3.2 数据访问控件 …… 199
10.3.3 使用 Recordset 对象访问数据库 …… 200
10.4 使用 ADO 控件访问数据库 …… 204
10.4.1 Visual Basic 数据库访问技术 …… 205
10.4.2 ADO 控件 …… 205
10.4.3 ADO 数据访问控件 …… 207
10.5 用编程方法引用 ADO 对象简介 …… 211
习题 10 …… 215

附录 实验 …… 217

实验 1 VB 环境的使用和简单应用程序设计 …… 218
实验 2 顺序结构与选择结构 …… 218
实验 3 循环结构 …… 219
实验 4 过程的使用 …… 220
实验 5 常用控件应用之一 …… 220
实验 6 常用控件应用之二 …… 222
实验 7 键盘与鼠标 …… 222
实验 8 菜单 …… 223
实验 9 工具栏 …… 223
实验 10 文件 …… 224
实验 11 图形处理与多媒体应用 …… 224
*实验 12 数据库的创建与数据控件 …… 225

参考文献 …… 227

第1章　可视化编程与VB开发环境

学习目标

了解可视化编程的基本概念包括类、对象、属性、事件与方法等。掌握VB的开发环境及其使用方法，通过几个实例初步学会使用VB进行程序设计的步骤。理解工程的含义及对工程的管理方法，掌握在VB的使用中获得帮助的方法。

可视化编程是目前广泛流行的程序开发方法。本章主要介绍可视化编程的概念及其开发平台，面向对象程序设计的基本概念，Visual Basic中的类、对象、属性、方法、事件和事件过程等基本概念，以及Visual Basic的功能特点，以便使读者对Visual Basic有一个总体认识。通过可视化编程概念的学习，使读者初步认识VB程序的开发环境和开发模式，初步学会设计VB程序。

【案例一】 欢迎进入奇妙的VB世界。

这是用VB开发的一个非常简单的界面，程序运行后显示如图1-1所示的窗口，单击“确定”按钮后，在文本框中显示“欢迎进入奇妙的VB世界”，如图1-2所示。

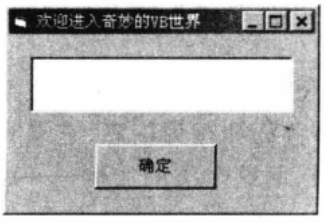


图1-1　运行时的初始画面

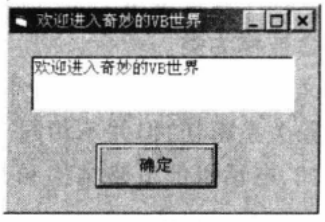


图1-2　结果画面

1.1　可视化编程的概念

随着图形用户界面(Graphic User Interface，GUI)操作系统Windows的广泛使用，用户不再需要记忆各种命令，而是通过鼠标对操作对象的单击、双击或拖放等即可轻松地完成相应的操作。

可视化(Visual)编程技术使得编程人员在编写应用程序时，既不需要编写大量的代码来描述组成用户界面的各种对象的外观和位置，也不需要通过反复调试、运行来调整它们，而只要把由开发工具提供的各种对象拖放到屏幕上并直接调整位置、大小、颜色和字体等即可。程序运行时所见的结果就是设计时调整的结果，也就是常说的“所见即所得(What You See Is What You Get，WYSIWYG)”。

由此可见，可视化编程犹如搭积木式的游戏，大大提高了应用程序的开发效率，降低了程序员的劳动强度。许多高级语言都相继推出了其可视化版本，这类开发工具一经推出便深受编程

人员的喜爱。本书中的 Visual Basic 就是一种典型的可视化编程开发工具,除此之外,像 FoxPro、C 和 Java 等也都有它们的可视化版本。

1.1.1 面向对象的程序设计

面向对象的程序设计方法与编程技术不同于传统的面向过程的程序设计。程序设计人员在进行面向对象的程序设计时,不再是单纯地从代码的第一行编到最后一行,而是通过考虑如何创建对象,利用对象来简化程序设计,提供代码的可重用性。

对象可以是应用程序中一个自包含组件,一方面具有私有的功能,供自己使用;另一方面又提供公用的功能,供其他用户使用。对象包含数据及其代码,这比传统编写代码的方法更容易维护。

在 Visual Basic 中,应用面向对象的程序设计(OOP)方法,把程序代码和数据封装起来作为一个对象,大部分对象都是可视的。程序员在设计程序时只需用现有工具根据界面设计的要求,直接在屏幕上"画"出窗口、菜单、按钮和滚动条等不同类型的对象,并依次为这些对象设置属性。程序员编程工作的重点仅限于编写相关对象要完成的功能代码,因而程序设计的效率可大大提高。

1.1.2 类、对象、属性与方法

1. 类

面向对象程序设计基本上是一种包装代码,代码可以重用而且维护起来很容易,其中最主要的包装概念被称为类。

类是定义了对象特征以及对象外观和行为的模板,可以使用"类设计器"来创建类。

类和对象关系密切,但并不相同。类包含了有关对象的特征和行为信息,它是对象的蓝图和框架。例如,电话机的电路结构和设计布局可以是一个类,而这个类的实例——对象便是一部电话机。

2. 对象

对象是类的一个实例,包括了数据及其代码。例如,一个对话框、一个命令按钮或一个文本框均可视为一个对象。

在可视化语言编程环境中,将用户界面上出现的元素,例如命令按钮、文本框和列表框等都看成对象。对象是 VB 程序设计的核心,VB 已经设计好了各种对象,程序员可以直接使用这些对象。

对象具有 3 个要素:属性、方法和事件。

3. 属性

为了便于操作和控制对象,在建立对象时,赋予了它们许多属性。每个对象都有属性,例如,一部电话机有颜色和大小;当把一部电话机放在办公室中,它又有了一定的位置;而它的听筒也有拿起和挂上两种状态。这些属性体现了该对象的外观和对事件的响应能力等特性,即属性是对象的性质或是描述对象的数据。改变对象的属性,便可控制其在程序中的作用。

属性的类型因对象而异,比如树木具有种类、形状、颜色以及高度等可见属性;还有一些不可见的属性,如寿命、材质等,所有的树木都可以具有这些属性,但不同的树木个体其属性的值

却各不相同。

对象的属性可以进行设置。有些属性可以在设计时通过属性窗口来定义，如影响一个控件在运行时是否可见的 Visible 属性、对象的名称、标题等；有些属性可在运行时通过编写代码来设置，如文本内容、菜单条目等。

4. 方法

属性是描述对象的数据。方法是让对象实施一个动作或执行一项任务的途径，即方法告诉对象应处理的事情。每一个对象都包含对数据进行操作的代码段，这段代码就是对象能够执行的一个操作，即方法。例如，列表框有 AddItem（增加项目）、RemoveItem（删除项目）和 Clear（清除所有项目）等方法来维护其列表中的内容。

1.1.3 事件及事件过程

事件是一种预先定义好的特定动作，由用户或系统触发。

例如，当用户在 Windows 桌面上用鼠标单击“开始”按钮，单击此按钮的动作就是一个事件。每当这个事件发生时，系统将弹出“开始”菜单，让用户选择，再根据选择触发下一个事件。

对于每个对象，都规定了相应的响应事件，如鼠标的单击、键盘上的按键、对象内容的更改等。

事件是对象在应用程序运行时所产生的事情，即生成的对象所要完成的任务，如用户单击一个命令按钮就是一个事件。每发生一次事件，将发送一条消息至操作系统。操作系统处理该消息并广播给其他窗口。然后，每一个窗口才能根据自身处理该条消息的指令而采取适当的操作（例如，当窗口解除了其他窗口的覆盖时，重显自身窗口）。

在传统的或“过程化”的应用程序中，应用程序自身控制了执行哪一部分代码和按何种顺序执行代码。而在事件驱动的应用程序中，事件可以由用户操作触发，也可以由来自操作系统或其他应用程序的消息触发，甚至由应用程序本身的消息触发；代码也不是按照预定的路径执行，是在响应不同事件时执行不同的代码片段；这些事件的顺序决定了代码执行的顺序，因此应用程序每次运行时执行代码的路径都是不同的。

代码在执行中也可以触发事件。例如，在程序中改变文本框中的文本将触发文本框的 Change 事件。如果 Change 事件中包含有代码，则将导致该代码的执行。

总之，可以把事件看做是一个响应对象行为的动作，事件发生时可以编写代码进行处理。每个事件都与一段代码相关，与事件相关的代码称为“事件过程”。

1.2 可视化程序设计的开发平台

Windows 环境下有非常丰富的可视化开发工具，主要有 Borland（Inprise）公司的 Delphi、C ++ Builder，Microsoft 公司的 Visual Studio（其中包括 Visual C ++、Visual Basic、Visual FoxPro、Visual J ++ 等），其中，Visual Basic 由于继承了 BASIC 语言简单易学的优点，同时又增强了数据库及 Internet 的编程功能，因此已经成为开发 Windows 应用程序的主要工具。本书就以 Visual Basic 为平台介绍可视化编程的基本知识。

1.2.1 Visual Basic 开发平台简介

Visual Basic 有3种版本可供使用，每种版本都是为了满足特定的开发需求。

1. Visual Basic 学习版

Visual Basic 学习版是一个入门版本，可以让编程人员方便地建立 Windows 应用程序，具有建立 Windows 应用程序所需的全部工具。

2. Visual Basic 专业版

专业版向计算机专业人员提供了一套功能完整的开发工具，它包含了学习版的所有功能，还附加有 ActiveX 控件、Internet Information Server 应用程序设计器、集成数据工具和数据环境、Active Data Objects 以及动态 HTML 页面设计器。

3. Visual Basic 企业版

企业版允许专业人员以小组的形式来创建分布式应用程序。它包括专业版的所有特性，加上 Visual SourceSafe(一种版本控制系统)和 Automation and Component Manager(自动化和组件管理器)等工具。

1.2.2 Visual Basic 的功能与特点

Visual Basic 是由 Microsoft 公司于 1991 年推出的第一个可视化编程工具。它经历了 3.0、4.0、5.0 和 6.0 等版本，随着 .NET 的推出，Visual Basic 也进入了 .NET 时代。其中 Visual Basic 6.0 是 1999 年推出的一个更新版本，其使用非常广泛，至今还是学习面向对象程序设计的主要平台，也是本书编写的基础平台。其主要功能与特点如下。

① 具有面向对象的可视化设计工具。

② 事件驱动的程序运行机制。

③ 提供了易学易用的应用程序集成开发环境。在 Visual Basic 集成开发环境中，用户可设计界面、编写代码、调试程序，直至将应用程序编译成可执行文件在 Windows 中运行，使用户在友好的开发环境中工作。

④ 结构化的程序设计语言。Visual Basic 具有丰富的数据类型，它也是一个符合结构化程序设计思想的语言，而且简单易学。此外，作为一种程序设计语言，Visual Basic 中还有许多独到之处：

- 强大的数值和字符串处理功能。
- 丰富的图形指令，可方便地绘制各种图形。
- 提供静态和动态数组。
- 过程可递归调用，使程序更为简练。
- 支持随机文件访问和顺序文件访问。
- 具有完善的运行出错处理。

⑤ 支持多种数据库系统的访问。利用数据控件可访问的数据库系统有 Microsoft SQL Server、Microsoft Access、dBASE、Visual FoxPro 和 Paradox 等；也可访问 Microsoft Excel、Lotus 1-2-3 等多种电子表格。

⑥ 支持动态数据交换(DDE)、动态链接库(DLL)和对象链接与嵌入(OLE)技术。动态数据交换(DDE)的编程技术，使 Visual Basic 开发应用程序能与其他 Windows 应用程序之间建立数据通信。

Visual Basic 通过动态链接库(DLL)技术,在程序中可方便地调用 C 语言或汇编语言编写的函数,也可调用 Windows 应用程序接口(API)函数,以实现 SDK 所能实现的功能。

Visual Basic 的核心是其对对象的链接与嵌入(OLE)的支持,它是访问所有对象的一种方法。利用 OLE 技术,Visual Basic 将其他应用软件视为一个对象嵌入到 Visual Basic 应用程序,实现声音、图像和动画等多媒体功能。

⑦ 完备的联机帮助功能。Visual Basic 的帮助系统有机地与 Visual Studio 开发工具的帮助系统结合在 MSDN 中,其中包含有大量的知识库和示例,为读者的学习和使用提供了极大方便。

本书约定:

- Visual Basic 简称为 VB。
- 标记☞表示技巧,❗表示注意事项。

1.3 VB 开发环境初识

1.3.1 VB 的启动

启动 VB 可以通过选择〖开始〗→〖所有程序〗→〖Microsoft Visual Basic 6.0 中文版〗→〖Microsoft Visual Basic 6.0 中文版〗命令,可以启动 Visual Basic 6.0。初次启动时,将显示如图 1-3所示的"新建工程"对话框,要求用户选择工程的类型。

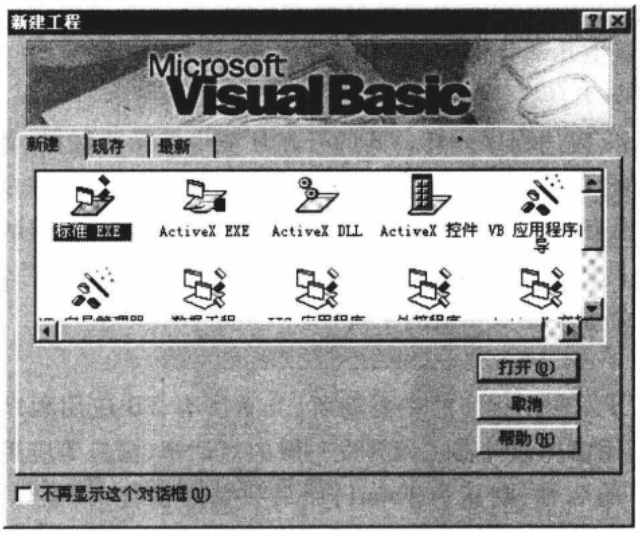


图 1-3 选择工程的类型

该对话框中有"新建"、"现存"和"最新"3 个选项卡,其中在"新建"选项卡中列出了 VB 可以建立应用程序的类型,如"标准 EXE"、"ActiveX EXE"、"VB 应用程序向导"等;在"现存"选项卡上,可选择并打开现存的工程;在"最新"选项卡上,可选择并打开最近编辑过的工程。选择"新建"选项卡中的"标准 EXE",将创建一个包含标准窗体的工程文件(即创建一个典型的 Windows 应用程序),进入如图 1-4 所示的集成开发环境。

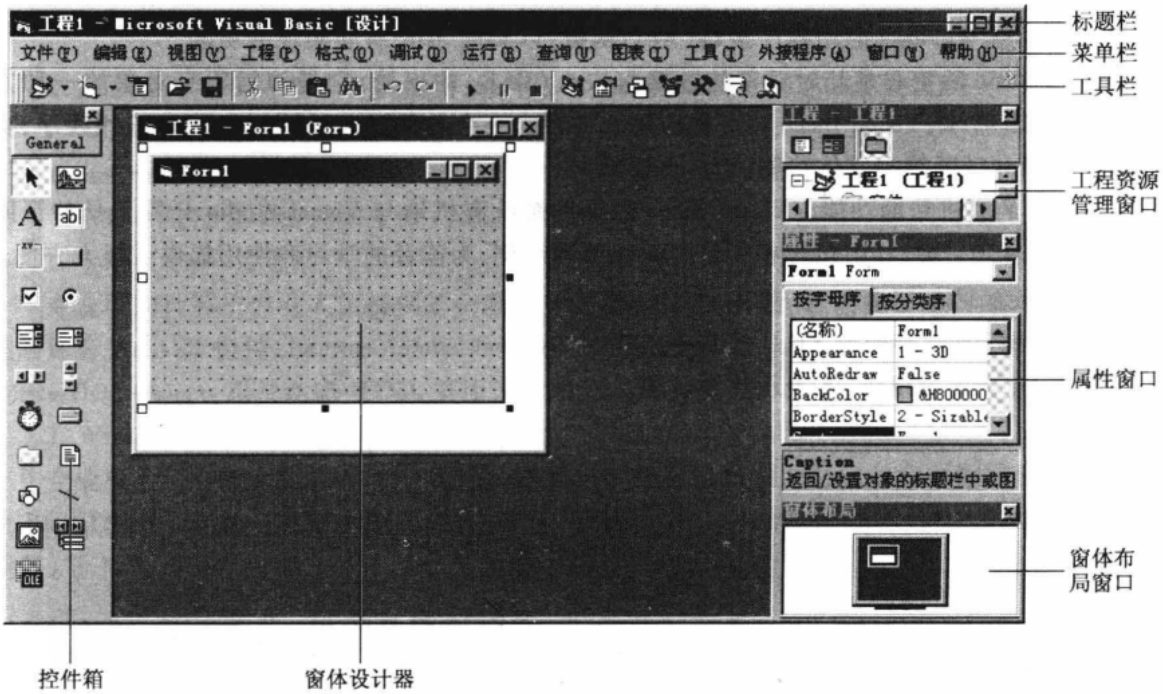


图 1-4　VB 的开发环境

1.3.2　开发环境的组成

VB 的开发环境由以下几部分构成。

1. 标题栏、菜单栏与工具栏

与绝大部分 Windows 应用程序一样，VB 的开发环境也包括标题栏、菜单栏和工具栏。标题栏中是当前开发的工程名称，默认为“工程 1”。菜单栏中集成了应用程序开发中所需要的各种菜单命令。VB 的工具栏有 4 种：“编辑”、“标准”、“窗体编辑器”和“调试”，默认情况下只显示“标准”工具栏，其他工具的显示可通过右击工具栏选择相应的工具栏名称即可，也可以选择〖视图〗→〖工具栏〗菜单命令来显示相应的工具栏。

2. 窗体设计器

窗体设计器是 VB 开发应用程序的主要场所，几乎所有 VB 应用程序的界面都是通过窗体设计器开发出来的。利用窗体设计器可以开发一般的对话框、窗口等应用程序的主要界面。

每个窗体都有唯一的名称，默认为 Form1，以后再新建窗体时，其默认名称分别为 Form2、Form3……一个应用程序至少应该包含一个窗体，当然，应用程序通常包含多个窗体。用户使用应用程序时，正是通过窗体来与程序进行交互的。

在窗体上有许多网格，主要用来在设计时对齐窗体上的各种对象，在运行时该网格并不可见。

3. 控件箱

控件箱如图 1-5 所示，控件箱又称控件工具箱或工具箱。选中控件箱中的任何一个工具，就可以在窗体设计器上绘制相应的控件。在控件箱中有 20 个以按钮形式表示的各种控件制作工具，这些控件是设计制作 VB 应用程序界面上各种对象的工具（因此也叫控件工具箱）。另外还有一个

指针按钮，尽管它不能制作控件，但必须使用它来选择、移动窗体及控件或者调整其大小。通过工具栏上的“工具箱”按钮或选择〖视图〗→〖工具箱〗菜单命令均可显示或隐藏控件箱。

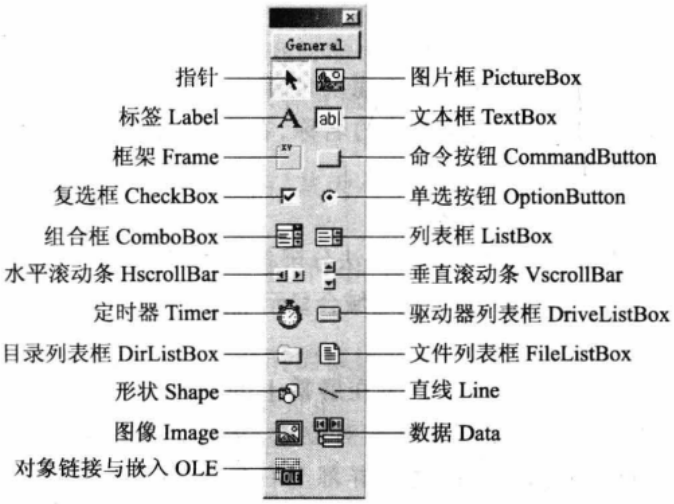


图 1-5 控件箱

控件箱中所显示的控件都称为内部控件，当然也可以添加其他非内部控件，如各种 ActiveX 控件。添加的具体方法请参见本书后面的描述。

4. 工程管理器窗口

工程管理器又称工程资源管理器，它以层次列表的方式管理当前工程文件（以 .vbp 为扩展名）或工程组文件（以 .vbg 为扩展名）中所包含的窗体文件（其扩展名为 .frm）、模块文件（以 .bas 为扩展名）和类模块文件（以 .cls 为扩展名）等文件。

5. 属性窗口

属性窗口如图 1-6 所示，用于显示和设置窗体或控件的所有属性。属性窗口有两个选项卡，一个按字母顺序显示出控件的所有属性，另一个按类别组织这些属性。属性窗口由 3 部分构成。



图 1-6 属性窗口

(1) 对象列表框

对象列表框中包含了当前窗体中所有对象名称列表,利用该下拉列表框可以选择需要设置属性的对象。

(2) 属性列表框

属性列表框是进行窗体设计中使用最为重要的部分,在其中列出了当前对象在设计时可以更改的属性名称及默认值。左边列出其属性的名称,右边则列出属性对应的值。每个属性都有一个默认值,对默认值可以进行修改。

(3) 属性含义说明

由于每个对象的属性很多,在设计时可能无法分辨各个属性的意义。属性含义说明正是为了解决这样的问题。当选定了某个属性名后,在该区域会显示所选属性的简要说明以帮助用户对其设置。

6. 窗体布局窗口

窗体布局窗口用于确定程序运行时窗体在屏幕上的初始位置。当一个应用程序中有多个窗体时更为有用,它也可以调整各个窗体的相对位置,其主要目的是使应用程序能在各个不同分辨率的屏幕上正常运行。在该窗口的模拟屏幕上右击会出现快捷菜单,其中包括"分辨率向导"、"启动位置"、"可连接的"和"隐藏"命令。利用该菜单或鼠标的拖动,可以方便直观地设置窗体运行时在屏幕上的位置。

除了以上所介绍的基本部分以外,在VB程序设计中还有一个重要的窗口,即代码窗口,它在VB启动时并没有出现在屏幕上。有关代码窗口的显示与使用方法在本书以后的章节中将有详细的叙述。

1.4 VB快速入门

利用VB开发应用程序与使用传统的高级语言开发应用程序的步骤完全不同,它充分体现了面向对象程序设计的特征。其基本过程是:

① 设计应用程序界面。

② 设置属性。

③ 编写代码。

④ 运行、调试。

下面通过对案例一的设计过程来说明利用VB开发应用程序的基本方法。

1.4.1 案例一:设计"欢迎进入奇妙的VB世界"

如图1-1、图1-2所示,程序界面由一个文本框和一个命令按钮组成,单击命令按钮,文本框中会出现"欢迎进入奇妙的VB世界"字样。

1. 设计界面

VB应用程序的界面是通过在窗体上设置各种各样的控件来实现的。通过VB控件箱,可以将所需要的控件添加到窗体上,以构成应用程序界面。因此,设计VB应用程序的第一步就是创建窗体。

(1) 创建窗体

启动VB后如果选择"标准EXE"会自动新建一个窗体。对于本实例程序,就可以直接在该

窗体上绘制控件了。操作步骤如下：

① 在工具箱内单击所需的控件工具按钮，如单击工具箱中的“文本框”工具ab|以便在窗体上绘制“文本框”控件。

② 将指针移到窗体 Form1 上，该指针形状变为“十”字形。

③ 拖动鼠标，画出大小合适的矩形框，释放鼠标按钮时，“文本框”控件便出现在窗体上。

(2) 调整控件大小

绘制命令按钮 Command1，控件添加后，其大小不一定合适，这样就需要调整控件的尺寸。控件大小的调整方法如下：

① 通过鼠标单击选中控件，被选定的控件上出现尺寸控点。尺寸控点是指出现在控件四周的 8 个深色小方块，用于调节控件的大小。

② 将鼠标指针定位于尺寸控点上，此时的鼠标变成双向箭头形状。

③ 拖动该尺寸控点直至所希望的大小为止。

④ 释放鼠标。

利用 4 个角上的尺寸控点可以同时调整控件的水平方向和垂直方向的大小，而 4 条边上的尺寸控点只能用于调整控件一个方向的大小。

在选定控件后，可用 Ctrl 键 + 方向键，每次移动控件一个网格单元。

通过以上操作，已生成了“欢迎进入奇妙的 VB 世界”应用程序的界面，如图 1-7 所示。

2. 设置属性

窗体及其所属的控件都有默认的属性，但默认的属性不一定符合程序的需要，因此常常要对部分属性重新进行设置。设置属性就是使用属性窗口或在运行阶段指定对象的属性。

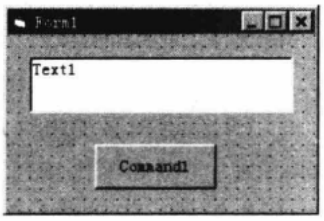


图 1-7 示例应用程序的界面

属性窗口位于主窗口的右边(见图 1-4)，由于不同的对象具有不同的属性，因此选中不同的对象，在右侧会出现相应的属性窗口。

如果在屏幕上没有属性窗口，则可以通过选择〖视图〗→〖属性窗口〗菜单命令或在工具栏中单击“属性窗口”按钮来显示。

在属性窗口中设置属性很简单，只要在属性列表中选定属性名称，然后在属性窗口的右列中输入或选择新的属性值即可。

有些属性有预定义的设置值清单。单击设置框右侧的下拉箭头，可以显示出这个清单；双击列表项，可以循环显示这个清单。

对本案例程序需要改变的属性及其设置值见表 1-1，其他属性则采用默认值。

表 1-1 案例一应用程序属性的设置

控 件 名	属 性 名	属性设置值
Form1	Caption	欢迎进入奇妙的 VB 世界
Text1	Text	
Command1	Caption	确定

3. 编写代码

经过前面的操作已经设计好了应用程序的界面，但这个程序如果运行将只会显示一个窗体，单击“确定”按钮时窗体不会有任何反应。这是因为程序中还缺少针对单击事件做出响应所需要的代码。编写代码就是在代码编辑器窗口（简称代码窗口）内编写对象的事件过程代码，如图 1-8 所示。

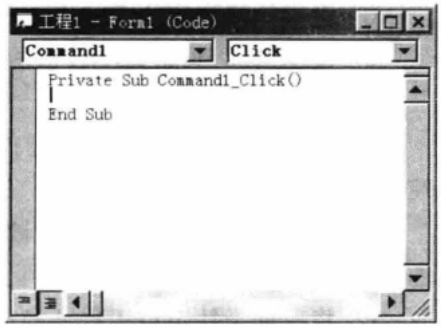


图 1-8　代码编辑器窗口

代码窗口是编写 VB 应用程序代码的地方。代码由语句、常量和变量声明等部分组成（有关 VB 的基本语法请参见第 4 章相关内容）。使用代码窗口，可以快速查看、编辑和调试 VB 应用程序代码的任何部分。

（1）打开代码窗口

代码窗口一般不显示在屏幕上，使用下列操作方法之一，可以打开代码窗口。

- 双击要编写代码的窗体或控件，此时会出现选定对象默认过程的代码窗口（见图 1-8）。
- 用鼠标右击窗体或控件选择“查看代码”命令，会出现代码窗口。

图 1-8 显示了在双击命令按钮控件后出现的代码窗口以及该命令的事件。代码窗口的上方包含有两个下拉列表框。

- 对象列表框：显示所选对象的名称。单击列表框右边的下拉箭头，显示与该窗体相关的所有对象的清单。
- 过程列表框：列出对象的过程或事件。该框显示出选定过程的名称。例如，在图 1-8 所示情况下，是 Click 事件，选取该框右边的向下箭头，即可显示出与此对象相关的全部事件。

（2）事件过程代码的创建

VB 应用程序的事件过程代码要放置在称为过程的代码块中（见图 1-8）。事件过程包含了事件发生（例如单击按钮）时要执行的代码。控件的事件过程是由控件的实际名称（Name 属性中所指定的）、下划线（_）和事件名称组合而成。

例如，单击一个名为 Command1 的命令按钮时调用的 Click 事件过程称为 Command1_Click() 事件过程。

创建事件过程代码的具体步骤如下。

① 打开代码窗口，在对象下拉列表框中，选定当前活动窗体中某一个对象名，如选定命令按钮 Command1。

② 在过程列表框中，选择指定对象的事件名称。此时 Click 过程已经被选定，因为它是命令按钮的默认过程。事件过程的模板就会显示在代码窗口中。在两条语句的中间输入如下的代码：

```
Text1. Text = "欢迎进入奇妙的 VB 世界"
```

该代码只简单地改变了控件 Text1 的 Text 属性，其含义为：当单击命令按钮时将在文本框 Text1 中显示“欢迎进入奇妙的 VB 世界”的字样。这个例子的语法采用了“对象名．属性名”的格式，其中 Text1 是对象名，Text 是属性名。

4. 运行应用程序

代码输入完成后就可以运行该示例程序了。运行 VB 应用程序的方法有如下 3 种：

- 选择〖运行〗→〖启动〗菜单命令。
- 单击工具栏中的“启动”按钮 ▶。
- 按 F5 键。

程序启动时的画面如图 1-1 所示。单击窗体上的“确定”按钮,文本框中就会显示“欢迎进入奇妙的 VB 世界”,如图 1-2 所示。要结束程序的运行只要单击当前窗口标题栏右端的“关闭”按钮即可。

5. 程序的保存与装入

在开发程序的过程中,为防止因各种意外情况导致工作丢失,要随时保存开发的程序,一般是在调试运行程序之前进行保存。

前已述及,一个 VB 应用程序包括了 4 类文件:工程文件、窗体文件、标准模块文件与类模块文件。因此,在保存工程之前,必须先保存窗体文件与标准模块文件(如果有的话),然后才能保存工程文件。本案例中只有窗体文件与工程文件。

保存文件的操作与一般的 Windows 应用程序相类似,也是要先选择保存位置,然后给文件命名。VB 中窗体文件的默认名为“Form1. frm”,可将其修改成合适的文件名;工程的默认文件名为“工程 1. vbp”,亦可修改其文件名。

VB 中默认的文件存放路径是 VB 的安装文件夹(“VB98”),由于 VB 应用程序所包含的文件比较多,所以一般将相关的文件放置在专门的文件夹下(可以事先建好)。

如果保存以后又进行了修改,也需要保存,此时的保存就不会再有“另存为文件”对话框了,除非选择了“工程另存为…”需要重新命名或改变保存位置。

程序保存后是以文件的形式存放在磁盘上的,如果下次还想继续修改,可以重新将该程序装入。

图 1-3 是启动 VB 时的界面,其中有 3 张选项卡。在“现存”选项卡中,用户通过“打开工程”对话框来打开一个工程;“最新”选项卡中(见图 1-9)显示的是最近用户所开发修改的工程,只要选中,单击“打开”按钮就可以打开相应的工程文件了。

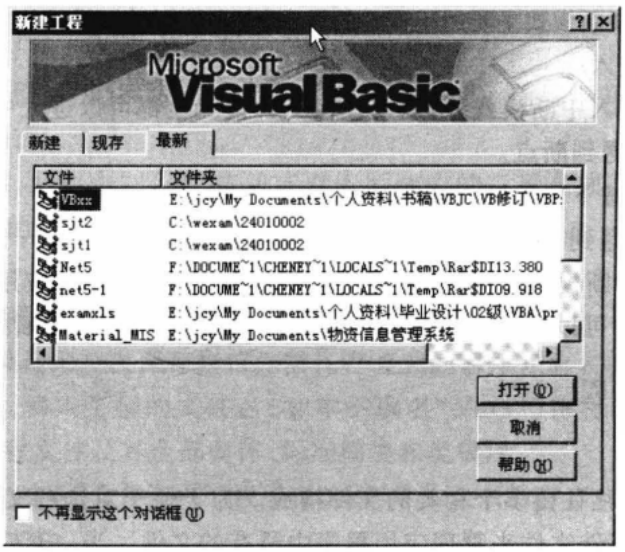


图 1-9 “最新”选项卡

1.4.2 VB的开发模式

VB应用程序的开发往往要经过设计、调试和运行3个基本步骤。为此VB提供了与此相适应的3种开发模式:设计模式、中断模式和运行模式。

在应用程序的开发过程中,任何时刻都要知道应用程序正处在哪种模式之下。在VB的标题栏上显示了当前所处的模式。

1. 设计模式

在VB中创建应用程序的绝大多数工作都是在设计模式下完成的。在设计模式时,可以设计窗体、绘制控件和编写代码,并使用属性窗口来设置或查看属性设置值。

尽管在设计模式下可以改变应用程序代码或重新设置控件的属性,但在设计模式下却看不到这些变更对应用程序运行产生的影响。要想观察设计改变后对程序运行的影响,必须将设计模式切换到运行模式。

2. 运行模式

当采用1.4.1节中的方法运行应用程序的方法运行一个程序时,即进入了VB的运行模式。在运行模式下,用户可以与应用程序进行交互,可观察应用程序的工作状况,可查看代码,但不能直接改动代码。

处于运行模式下的应用程序既可以单击工具栏上的"结束"按钮退回到设计模式,也可以单击"中断"按钮进入中断模式。

3. 中断模式

(1) 中断模式的基本概念

应用程序在开发环境下运行时,若暂时中止程序的执行将进入中断模式。在中断模式下,可以检查、调试、重置、单步执行或继续执行程序。

可根据需要随时进入中断模式以暂停应用程序的执行,并提供有关应用程序情况的快照。因为变量和属性设置值被保留下来,所以,可以分析应用程序的当前状态并修改其内容,这些修改将影响应用程序的运行。所以中断模式主要用于程序的调试,大多数调试工具只能在中断模式下使用。

(2) 中断模式的进入

除了上述方法可进入中断模式外,以下任何一种情况也都可以进入中断模式:

- 在执行程序时遇到断点。
- 在执行程序时按下快捷键Ctrl + Break。
- 在执行程序时遇到Stop语句或产生了某些错误。

当应用程序处于中断模式时,可以使用VB所提供的调试工具对应用程序进行各种调试(具体参见第4章),同时还可以随时重新启动应用程序、结束运行或从中断处继续运行等。

1.4.3 VB的工程

1. 工程概述

一个VB应用程序往往由多个相关的文件构成。为了便于对这些文件进行有效的管理,VB使用了称为"工程"的特殊文件来管理应用程序中所有的文件。当一套应用程序开发完成后,便可以对工程进行编译以创建一个可执行文件。一个完整的工程往往包括:

① 跟踪所有部件的工程文件(.vbp)。

② 窗体文件(.frm)。

③ 窗体所对应的二进制数据文件(.frx)。它包含了窗体上控件的属性数据。含有二进制属性(如图片或图标)的任何 .frm 文件都是自动产生的,且不可编辑。

④ 标准模块所对应的一个文件(其文件扩展名为 .bas),该文件是可选项。

工程文件是与该工程有关的全部文件的清单,工程文件还引用了工程中所使用的 ActiveX 控件和可插入对象。每次保存工程时,这些信息都要随之被更新。

"工程管理器"窗口又称为"工程资源管理器"窗口或"工程"窗口。当创建文件或在工程中添加、删除文件时,VB 便会在"工程管理器"窗口(见图 1-10)中反映出所发生的变化,该窗口包含此工程中当前文件的列表。

双击一个现存工程的图标,或选择〖文件〗→〖打开工程〗菜单命令,或拖动该文件至"工程管理器"窗口内,即可打开这个现存的工程文件。



图 1-10 "工程资源管理器"窗口

2. 工程的管理

(1) 新建工程

① 选择〖文件〗→〖新建工程〗菜单命令,出现相应的"新建工程"对话框,如图 1-3 所示。

② 选择一个工程类型后,单击"确定"按钮或双击所选中的工程类型,VB 便创建了一个新工程。

当选择了〖新建工程〗菜单命令后,VB 会关闭当前的工程,并提示用户保存已修改过的文件,然后才启动"新建工程"对话框,即可以从"新建工程"对话框选定一个工程类,VB 再创建一个带有单个新文件的新工程。

(2) 打开工程

① 选择〖文件〗→〖打开工程〗菜单命令。

② VB 关闭当前工程,并提示用户保存所有做过的改动。随后,VB 启动"打开工程"对话框,以便打开一个现有工程。

③ 选中一个工程名后,单击"打开"按钮或双击所选中的工程名称。此时,VB 就打开一个包含其工程文件中列出的所有窗体、模块和 ActiveX 控件的现有工程。

(3) 保存工程

保存工程的方法有两种:

- 直接保存。如果是第一次保存则首先需保存有关的窗体并命名(如本例取名"frmExam_2_1"),然后再保存工程,输入工程的文件名(如本例取名"Vbxx")即可。如果是再次保存,VB 只是更新当前工程的工程文件及其全部窗体、标准模块和类模块。
- 另存工程文件。选择〖文件〗→〖工程另存为〗菜单命令来另存当前工程的工程文件。

(4) 向工程中添加文件

可以通过选择〖工程〗→〖添加文件〗菜单命令将现存的相关文件添加到当前的工程文件中。

要注意的是，在工程中添加文件，并不是复制该文件，而是简单地将对该现存文件的引用纳入工程中。因此，若更改文件并保存它，则此变化会影响到其他引用该文件的任何工程文件。如想改变文件而不影响其他工程，则应在工程管理器里选中该文件后，选择〖文件〗→〖另存为〗菜单命令，再以另一个新名字保存并引用此文件。

☞ 可以从Windows的资源管理器或“我的电脑”拖动文件并放入“工程”窗口内，即可将它们添加到一个工程中；也可以拖动.ocx文件至工具箱内，即添加了新的控件。

(5) 从工程中删除文件

如果某些文件已经不再需要，可以在“工程资源管理器”窗口中选定要删除的文件，然后再选择〖工程〗→〖移除文件〗菜单命令(或右击该文件对象，在快捷菜单中选择“移除”菜单项)，则该文件会从工程里被删除掉，但它仍然存在于磁盘上，只是从此工程中移走了。如果需要，则可恢复该工程文件。

❗ 如果从“工程资源管理器”内的工程中删除了文件，则VB在保存该工程时便可对其进行更新。但是若在VB之外删除了被工程引用的一个文件(即删除了一个工程中的文件)，则VB无法将此信息进行更新；所以，当删除文件后再打开此工程时，VB会显示错误信息，警告有一个文件丢失了。

另外，还可以在工程中添加“附加工程”或删除一个工程，由于一般使用较少，故不再介绍。

3. 启动窗体的设置

一个应用程序中通常要包含多个窗体，程序运行时最先运行的窗体称为“启动”窗体。可以将工程中任一个窗体设置为启动窗体，方法是：选择〖工程〗→〖xxx属性…〗菜单命令(xxx为当前工程的名称)，打开工程属性对话框，然后在“通用”选项卡的“启动对象”下拉列表框中选择一个已经存在的窗体作为启动窗体，再单击“确定”按钮。

❗ 在学习VB的过程中，需要学习设计许多窗体。每个窗体设计完成后，在运行前都要进行启动窗体的设置，否则将无法运行所设计的窗体，除非工程中只有一个窗体。

1.5 获取帮助

在VB应用程序的开发过程中总会遇到这样或那样的问题(如对某些属性的设置不熟、忘记了某些命令的语法等)，解决这些问题除了查阅相关的资料以外，参考VB所提供的联机帮助是最好的选择。

从VB 6.0开始将其帮助系统集成到了Microsoft Developer Network Library(MSDN)中，因此要使用VB的联机帮助必须先安装MSDN。只有在MSDN安装完成后，才能使用VB的联机帮助系统。

1.5.1 使用联机帮助

在任何情况下，按F1键或选择〖帮助〗→〖内容〗菜单命令均可启动VB的帮助系统，所不同

的是 F1 键提供了 Windows 系列软件通用的上下文敏感帮助功能。VB 帮助系统的主界面如图 1-11所示。



图 1-11 Visual Basic 联机帮助系统

1. 通过目录定位主题

单击左窗格中的“目录”选项卡即可查看帮助主题。

① 展开或收缩节点。在目录中单击节点前的加号(+)可展开此节点，然后双击要查看的项目。单击节点前的减号(-)可收缩此节点。

可使用上下方向键在垂直方向移动来定位节点，使用左右方向键可展开和收缩节点；使用工具栏上的“前进”和“后退”按钮来显示过去查看过的信息。

② 单击目录中的主题，可浏览该主题。

2. 通过索引查找信息

使用索引就像使用一本书的目录一样，可快速找到特定的信息。MSDN 的“索引”选项卡包含了一个关键字列表，这些关键字与众多的 MSDN Library 主题相关联。使用索引定位主题的方法是：

① 单击左窗格中的“索引”选项卡，然后输入或选择一个与所需查找信息有关的关键字，如“Caption 属性”。

② 选中关键字后，单击“显示”按钮(或按 Enter 键)。

③ 在查获的主题列表中选择所需的主题，然后单击“显示”按钮(或双击选中的主题)，将显示如图 1-12 所示的信息。



图 1-12　Visual Basic 联机帮助中的索引

1.5.2　从 Internet 上获得帮助

在 MSDN Library 中，有众多的内部链接以及与外部网站的链接。MSDN 查阅器使用了 Internet Explorer 浏览器的引擎，其默认的设置为脱机模式，但当激活了与外部网站的连接后就会切换到联机模式，与 Internet 上相关的网站建立连接。

除此之外，Internet 上有众多用于学习和交流的 VB 站点，包括微软官方站点。通过这些站点可以与世界各地的 VB 爱好者互相学习和交流，获取软件开发方面的资料。充分利用好 Internet，将使读者受益匪浅。

习　题　1

1. 试列举在 Windows 应用程序中常见的对象。
2. 可视化编程最主要的特点是什么？
3. 能进行可视化编程的语言有哪些？
4. 什么是对象、属性、方法和事件？试举例说明。
5. 对象、属性、方法和事件的关系是什么？试举例说明。
6. VB 6.0 的主要特点是什么？
7. VB 的开发环境由哪几部分组成？各部分的作用如何？
8. 使用 VB 进行程序设计有哪些步骤？
9. 如何创建、打开和保存 VB 文件？
10. 在进行一个应用程序开发时，会产生哪些文件？这些文件的作用是什么？
11. 如何调整窗体上控件的大小与位置？
12. 对控件的属性设置一般有哪几种方法？如何实现？
13. 运行应用程序的方法有几种？
14. 试简要说明 VB 的开发模式。

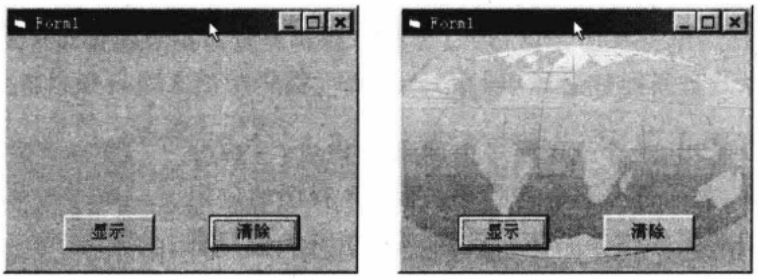
15. 如何才能进入中断模式?

16. 一个 VB 工程主要包括哪几类文件? 这些文件的扩展名是什么?

17. 在 VB 中能同时打开两个工程吗? 为什么?

18. 在进行 VB 程序设计时,可以通过哪些途径获取帮助信息?

19. 试设计一个简单的程序,要求如下:在窗体上设置两个按钮,一个按钮的标题为“显示”,另一个按钮的标题为“清除”。当单击“显示”按钮时,在窗体上显示一幅地球的图片;单击“清除”按钮时,将清除该图片。程序运行的结果如图 1-13 所示。



(a) 单击“清除”按钮 (b) 单击“显示”按钮

图 1-13 程序运行结果

提示:

(1) 图片文件在“C:\Program Files\Microsoft Office\Office\Bitmaps\Styles\Globe. wmf”。

(2) 加载图片文件用 LoadPicture 命令。

第2章　VB 简单应用程序的设计

学习目标

掌握窗体与控件常用属性的含义与设置、窗体的布局与调整，熟练掌握3种基本控件（标签、命令按钮与文本框）的设置与应用，学会使用代码窗口编写简单的代码，同时能设计简单的应用程序。

由前一章可以看出，一个 VB 程序由两部分构成：一部分是程序的应用界面，另一部分是程序代码。而 VB 程序的应用界面离不开控件的使用和相关属性的设置，VB 程序代码与事件及事件过程相关。VB 程序的应用界面是通过程序代码将界面中的各个对象有机地结合起来，从而实现程序所需要的功能。

【案例二】 简单的记事本。

Windows 中的“记事本”程序大家都用过，它具有简单的文字编辑处理功能，可以完成诸如复制、剪切和粘贴等基本功能。如图 2-1 所示的是简化的记事本，用 VB 很容易实现。

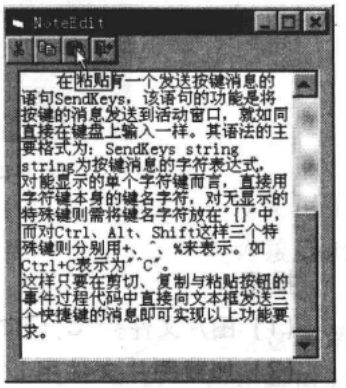


图 2-1　简单记事本

2.1　VB 中的对象

从案例一的制作过程可以看出，使用工具箱中的控件可以在窗体上设计程序界面。实际上，窗体与控件就是 VB 中定义好的标准对象，这些对象都是 VB 系统设计好后，提供给用户使用的，其移动、缩放等操作也是由系统规定的，但非常直观、方便。除了窗体与控件外，VB 还有如打印机、剪贴板、屏幕等其他对象。实际上，工具箱中所提供的是空对象（可以将其看成“类”），当将其放置到窗体上后，就形成了类的实例——对象，然后不仅可以调整其外观、位置，而且还可以设置其属性，控制其事件过程等。

2.1.1　对象属性及其设置

任何对象都有不同的属性，这些属性决定了对象的特征，如一个按钮具有 Caption（标题）、Name（名称）和 Color（颜色）等属性。所以，当在窗体上生成一个对象后，都需要对这个对象的相关属性进行设置，以满足程序的应用要求。

属性的设置主要有两种方法：一种是静态设置（设计时设置），另一种是动态设置（运行时设置）。

静态设置就是如第 1 章所介绍的那样，在窗体上选定某一个对象后，在属性窗口中即会出

现该对象的全部属性,然后可根据实际的需要对某些属性进行设置。这种设置一般是“所见即所得”的,常用来设置一些在程序运行期间一直不需要改变的属性,如对象的 Name(名称)属性等。

属性窗口中,各属性的设置根据属性的不同有 3 种方法:

① 直接输入新的属性值,这些值必须由用户来确定,如文本框的 Text 属性。这一类属性后面显示的就是一个文本框,单击属性名称后,即可在该文本框中输入属性值。

② 选择属性值。有些属性只有几种可能的值,不能由用户来输入,如窗体的 BorderStyle(边界类型)属性。这类属性在选定后,其属性窗口对应该属性的右边会出现下拉箭头。

③ 利用对话框来设置属性值。这类属性在选定后,右边会出现对话框标记按钮(…)。利用对话框可以对该属性进行多方面的设置。如 Font(字体)属性对话框打开后,可以像在 Word 中设置字体那样对对象的字体进行设置。

动态设置是指在程序中通过赋值语句来设置对象的属性,其一般格式如下:

对象名.属性名称=属性值(表达式)

例如,在 1.4 节中的“Text1.Text ="欢迎进入奇妙的 VB 世界""即采用这种格式。其中,Text1 为窗体上的一个文本框对象,Text 为该对象的一个属性(文本框中的内容),赋值号右边的是送入文本框内容的属性值。因此,只要能执行这一语句,就可以将文本框 Text1 中的内容设置为“欢迎进入奇妙的 VB 世界”。

动态设置在 VB 应用程序设计中应用非常广泛,凡在程序运行中需要改变的属性都必须采用这样的方法进行设置。至于在何种情况下设置,通过何种方式设置,就需要根据事件驱动的机制来确定。

2.1.2 对象事件与方法

VB 是基于事件驱动机制的面向对象编程语言,在这种机制下开发应用程序时,并不需要像传统的面向过程语言那样编写非常庞大的程序代码,而只需要建立一个由若干个微小程序组成的应用程序,这些微小的程序都可以由用户操作启动的事件来激发。

VB 中有许多这样的事件,这些事件都是由 VB 预先设置好的、能够被对象所识别的动作。常用的事件如 Click(单击)、DblClick(双击)、Load(装入)、MouseMove(鼠标移动)和 Change(改变)等。

事件总是与对象相联系的,不同的对象所能识别的事件也不同,但也有许多事件是许多对象所共有的,如 Click 事件。

对象在响应事件后一般总需要执行一些操作,这些操作就可以通过事件过程来实现,例如单击一个“打开”按钮,触发了该按钮的 Click 事件,随即执行该事件过程中的代码“Myfrm.Show”,这样就可以打开另一个窗口。VB 的事件过程都有一个固定的模板,其基本格式如下:

```
Private Sub 对象名称_事件名称([参数表])
    …
    '事件过程代码
    …
End Sub
```

其中,“对象名称”是该对象的 Name 属性;“事件名称”则是由 VB 预先定义好的赋予该对象的事件,这个事件该对象肯定能识别。有些事件是可以带参数的,参数的个数与形式都是 VB 已规定

的，用户只能引用这些参数。一个 VB 对象可以识别哪些事件并不需要用户来操心，只要在窗体上（实际上包括窗体本身）建立一个对象后，VB 就能自动确定与该对象所匹配的事件列表以供用户在编程时选择。

当然，由于一个对象可以识别的事件很多，在实际开发应用程序时并不需要对每个事件都编写事件代码，而只需要根据实际需要对极少的几个事件编写事件过程就可以了。

在传统的结构化程序设计语言中，过程与函数通常作为构成程序的主要部件。而在面向对象程序设计中则引入了称为“方法”的特殊过程或函数，它与传统的函数或过程是类似的，也有一个名字，也可以接收各种参数，也可以返回某些结果等，但是，它有一个明显的不同是：任何方法总是属于某个对象，就像属性、事件是对象的一部分一样，方法也是构成对象的重要组成部分。因此，方法的调用就不同于传统的过程或函数调用，其格式如下：

对象名称．方法名称［参数表］

如

Myfrm. Show

就是调用名为“Myfrm”窗体的 Show 方法，将该窗体在屏幕上显示出来。

与事件一样，VB 为每个对象都准备了许多方法，用于实现不同的功能，这些方法将在以后的学习中陆续了解。

为了便于区分窗体对象与一般的对象，以后对工具箱中所提供的各种空对象以及在窗体上生成的各种实际对象统一称为控件，窗体对象称为窗体。

2.2　窗体与控件概述

2.2.1　窗体及应用

在利用 VB 进行应用程序开发时，窗体相当于一块“画布”，可以在这块画布上画出各种各样的控件，继而通过对这些控件的事件进行编程，就可以开发出相应的应用程序。所以，窗体是 VB 应用程序建立的基础。

1. 窗体的结构与常用属性

VB 的窗体结构和特性与 Windows 的窗口十分类似。在设计阶段称为窗体，而当 VB 程序设计完成进入运行时也可以称为窗口或对话框。VB 窗体设计窗口如图 2-2 所示。

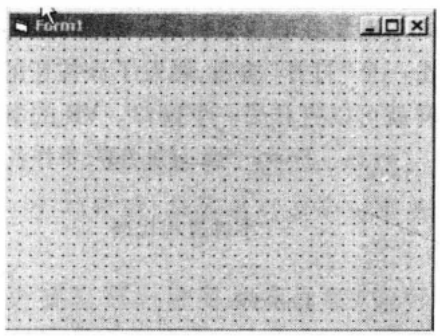


图 2-2　窗体设计窗口

窗体上也具有系统菜单、标题栏和最大化、最小化按钮，它们分别对应 ControlBox、Caption、MaxButton、MinButton 对象。

窗体的属性决定了窗体的外观与操作，其典型属性简介如下。

（1）Caption

窗体的标题属性，为显示在窗体标题栏中的文字。当

新建窗体时,其默认的 Caption 属性值为 Form1、Form2、Form3……用户可以将其修改为指定的名称。

(2) Name

该属性来定义窗体对象的名称。Name 属性是所有对象都具有的属性,它决定了对象在创建时所取的名称,当需要在代码中引用该对象时,必须给出对象的名称,这个名称不会出现在窗体上。对象的名称一旦设定,一般不要修改。

(3) MaxButton(MinButton)

这两个属性是用来控制窗体是否显示最大(最小)化按钮的。它们是逻辑量,如取 True 则显示,否则不显示。当不希望用户在操作中改变窗口的大小时,通常将其设置为 False。

(4) ControlBox

每个窗口在其左上角都有一个控制菜单框(系统菜单框),用来控制窗口的大小、位置等。该属性就是用来控制该菜单框是否显示的,同样为逻辑量,意义与 MaxButton 属性相同。

(5) BackColor

窗体的默认背景颜色为灰色,如果要改变为其他颜色,可以使用该属性。当在属性窗口中修改该属性时,会显示一个调色板来供用户选择;如果通过动态的方式来设置则通常使用 RGB()函数或颜色常数来产生颜色,有关 RGB()函数在后面会学习到。

颜色用一个十六进制数表示,格式是在两个 & 之间包含一个以 H 开头的 8 位十六进制数,如:&H8000000F&。

(6) BorderStyle

BorderStyle 用来确定窗体边框的类型,它有 6 种不同的选择,如表 2-1 所示。

表 2-1 窗体边框类型

设 置 值	含 义
0 - None	无边框窗体
1 - Fixed Single	固定单边框窗体,可以包含控制菜单框、标题栏、最大(最小)化按钮,但窗体的大小只能通过最大(最小)化按钮来改变
2 - Sizable	(默认值)窗体大小可变,有标准的双线边界,与标准的 Windows 窗口相同
3 - Fixed Dialog	固定对话框,可以包含控制菜单框与标题栏,但没有最大(最小)化按钮,窗体大小在设计时确定,运行时不能调整,通常用于设计对话框
4 - Fixed ToolWindow	固定工具窗口,窗体大小不变,只显示关闭按钮
5 - Sizable ToolWindow	可变大小的工具窗口,其余同上

(7) Height(Width、Left、Top)

确定窗体大小与位置的有 4 个属性,分别为 Height(高)、Width(宽)、Left(窗体到屏幕左边界的距离)和 Top(窗体到屏幕上边界的距离)。

VB 中所有涉及尺寸的单位都为 twip(缇),1 twip 是打印机的一磅的 1/20(1 440 twip = 1 in;

567 twip = 1 cm)。

除了以上所介绍的属性外,窗体还有许多其他的属性,这里就不一一介绍了,使用时可以参考联机手册或其他参考资料。

VB中不同的对象有许多属性的名称及含义是非常类似的,学习时要注意举一反三。

2. 窗体的常用事件

与窗体有关的事件很多,以下是几种典型的事件。

(1) Click事件

Click事件是VB中使用最为广泛的一个事件,几乎所有的可见对象都有此事件。程序运行后,单击窗口中某空白位置时,将触发该事件Form_Click()。如果单击的位置上有其他对象,则只能触发该对象的Click事件,而不是窗体的单击事件,从而调用的是对象的单击事件过程。

(2) DblClick事件

这是一个双击事件过程,双击实际上触发了两个事件:第一次单击鼠标时触发的是Click事件;第二次才触发DblClick事件。因此,在编写DblClick事件过程时,除非特别需要,一般不要编写Click事件过程。

(3) Load事件

Load事件主要用来在启动程序时对窗体相关的属性和变量进行初始化。这个事件是自动触发的,只要运行窗体程序就会触发该事件。

(4) Unload事件

当从内存中清除一个窗体(关闭窗体或执行Unload语句)时触发该事件。

(5) Paint事件

当窗体被移动或放大之后,或在一个覆盖该对象的窗体被移开之后,该对象部分或全部暴露时,触发此事件。

2.2.2　控件与焦点

一个应用程序的界面由两部分构成:窗体(Form)及其上的各种控件。控件不能脱离窗体而单独存在。VB中的控件(Controls)是用户界面上常见的对象,如命令按钮、文本框和列表框等。每个控件都有许多属性。对控件的属性可以进行设置,属性是用来定义控件的特征或某一方面行为的,是描述对象数据的。对属性的设置一般有两种方法:一是在设计时通过属性窗口来定义,如影响一个控件在运行时是否可见的Visible属性、用于在代码中引用的控件名称、显示在程序界面上的控件标题等;二是在运行时通过编写代码来设置,如文本内容、菜单条目等。

在Windows环境下,任何时候仅有一个窗口、窗体或控件能接收鼠标单击或键盘输入。当对象的标题或标题栏被突出显示时,说明该对象"具有焦点"。

当对象具有焦点时,才可接收用户的输入。在Windows环境下任一时刻可运行多个应用程序,但只有具有焦点的应用程序才有活动标题栏,才能接收用户输入。如在有几个TextBox(文本框)的VB窗体中,只有具有焦点的TextBox才能接收由键盘输入的文本。

要使对象获得焦点可以使用下列方法：

- 运行时选择对象。
- 运行时用访问键选择对象。
- 在代码中用 SetFocus 方法。

有些对象，它是否具有焦点是可以看出来的。例如，当命令按钮具有焦点时，标题周围将出现虚框，按钮本身也将突出显示，如图 2-3 所示。文本框具有焦点时，在文本框内会出现闪烁的光标。

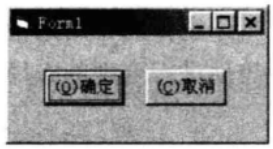


图 2-3 具有焦点的按钮

只有当对象的 Enabled 和 Visible 属性为 True 时，它才具有焦点。Enabled 属性允许对象响应由用户触发的事件，如键盘和鼠标事件。Visible 属性决定了对象在屏幕上是否可见。

只有包含任何可接收焦点的控件的窗体，才能接收焦点。

除标签、框架、计时器、菜单之外的控件都可以接收焦点。

2.2.3 控件的分类

VB 中有许多控件，大致可以将其分为 3 类。

1. 内部控件

VB 的内部控件也称为标准控件，如 CommandButton 和 TextBox 控件。这些控件总是出现在工具箱中，不像 ActiveX 控件和可插入对象那样需要时必须添加到工具箱中，也可以从工具箱中删除。

VB 的标准控件分为：按钮（如 CommandButton）、显示和输入文本的控件（如标签和文本框控件）、为用户提供选择的控件（如复选框、选项按钮、列表框、组合框和滚动条控件）、显示图片和图形的控件（如图像、形状和直线控件）、附加控件（如其他标准 VB 控件）等，如图 1-5 所示。

2. ActiveX 控件

ActiveX 控件是扩展名为 .ocx 的独立文件，其中包括各种版本 VB 提供的控件（如 DataCombo、DataList 控件）和仅在专业版和企业版中提供的控件（如 Listview、Toolbar、Animation 和 Tabbed Dialog），另外还有许多第三方软件开发商提供的 ActiveX 控件。利用 ActiveX 控件可以大大提高程序的开发效率。使用较多 ActiveX 控件的主要有以下几类。

- 公用控制类：如生成工具栏与状态栏的控件 ToolBar、ImageList、StatusBar 等。
- 通用对话框：利用 CommonDialog 控件可以生成多种标准 Windows 界面的对话框。
- 数据库管理类：这类控件较多，如 Adodc、DataGrid、DataList、DataCombo 等，利用这些控件可以很方便地开发数据库应用程。
- 多媒体类：如 Animation、MMControl 等。

3. 可插入的对象

可插入的对象如一个包含公司所有雇员列表的 Microsoft Excel 工作表对象或者一个包含某工程计划信息的 Microsoft Project 日历对象等。因为这些对象能添加到工具箱中，所以可把它们当做控件使用。其中一些对象还支持 OLE，使用这类控件就可在 VB 应用程序中编程控制另一

类应用程序的对象。

2.3　窗体的布局设计

窗体的布局是指如何向窗体上添加或删除控件，如何调整控件的大小及位置，如何对多个控件进行合理的布局等。

2.3.1　控件的添加/删除

1. 向窗体中添加控件

向窗体中加入控件，一般有两种操作方法。

- 在工具箱中单击所要添加的控件，然后将鼠标移动到窗体中，此时鼠标指针变为“十”字形，在需要加入控件的位置上按住鼠标左键拖动，直至所需要的尺寸，再释放鼠标左键，这时一个控件即被“画”到窗体上。

如果需要一次加入多个相同的控件，可在单击工具箱中的控件时先按住 Ctrl 键不放，这样就可在窗体上连续画出若干个相同的控件。

- 双击工具箱中所要的控件，该控件就会以默认大小自动出现在窗体的中心位置。随后，可以根据需要将该控件拖到合适的位置上。

2. 删除窗体上的控件

先单击选取将要删除的控件，再按 Delete 键即可。如果误删了某些控件，可按快捷键 Ctrl + Z 恢复。

2.3.2　调整控件的大小与位置

设计一个图形用户界面时，常常需要调整各控件的大小和彼此之间的相对位置。调整控件大小的方法是：首先用鼠标单击需调整的控件，在控件四周出现 8 个蓝色矩形小方块，称为尺寸控点，可通过尺寸控点调节控件的大小。

控件位置的调整有两种操作方法：一是用鼠标拖动对象，二是使用 Ctrl + 方向键移动。由于在设计窗体上存在网格，因此这两种移动都只能以网格为单位，这样也便于多个控件对齐。选择〖工具〗→〖选项〗菜单命令，在“选项”对话框的“通用”选项卡上，清除“对齐到网格”的选项，则控件的移动就不再受网格的限制。

☞ 按下快捷键 Ctrl + Shift，再用鼠标移动可以微调控件的位置。

❗ 有些控件运行时是不可见的，如定时器 Timer 控件、CommonDialog 控件，设置它们的位置和大小是无意义的。

2.3.3　窗体与控件的布局调整

1. 控件的布局调整

当窗体上安放了多个控件后，常需要将某一组控件进行布局调整，如将一组控件的尺寸设置成同样的大小，将一组控件进行对齐处理，等等。可以使用两种方法来进行控件的布局调整：

一是通过“格式”菜单，二是通过“窗体编辑器”工具栏。无论使用哪种方法，在调整前都必须先选定需要调整的控件。

多个控件的选定方法是：

- 使用 Shift 键配合鼠标的单击。
- 使用鼠标在窗体上拖出一个方框，在框内的控件将全部被选中。

当选中了需要调整的控件后，就可以使用“格式”菜单进行布局调整了。当然也可以通过工具栏进行。一般情况下，“窗体编辑器”工具栏并不显示在屏幕上，因此可选择〖视图〗→〖工具栏〗→〖窗体编辑器〗菜单命令来打开，如图 2-4 所示。

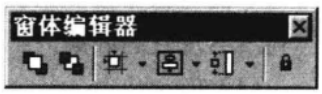


图 2-4 “窗体编辑器”工具栏

工具栏上有许多按钮，各个按钮的功能如表 2-2 所示。

表 2-2 布局调整按钮功能说明

按 钮	功 能 说 明
左边对齐	按最左边界对齐选定控件，当选定多个控件时可用
右边对齐	按最右边界对齐选定控件，当选定多个控件时可用
顶端对齐	按最上边界对齐选定控件，当选定多个控件时可用
底端对齐	按最下边界对齐选定控件，当选定多个控件时可用
垂直居中对齐	按照一垂直轴线对齐选定控件的中心，当选定多个控件时可用
水平居中对齐	按照一水平轴线对齐选定控件的中心，当选定多个控件时可用
相同宽度	把选定控件的宽度调整到与最宽控件的宽度相同
相同高度	把选定控件的高度调整到与最高控件的高度相同
相同大小	把选定控件的尺寸调整到最大控件的尺寸
水平居中	按照通过表单中心的垂直轴线对齐选定控件的中心
垂直居中	按照通过表单中心的水平轴线对齐选定控件的中心
置前	把选定控件放置到所有其他控件的前面
置后	把选定控件放置到所有其他控件的后面

2. 窗体布局调整

窗体布局调整是指决定程序运行时，窗体在屏幕上出现的位置。利用“窗体布局”窗口可以调整窗体的布局。如果在屏幕上没有“窗体布局”窗口，可选择〖视图〗→〖窗体布局窗口〗菜单命令或单击工具栏上按钮显示它。在“窗体布局”窗口中可以预览及定位窗体，就像它出现在应用程序中一样。使用快捷菜单中的〖分辨率向导〗命令，可以按不同的分辨率进行预览。

在设计时，“窗体布局”窗口用于可视地定位窗体，把环境中所有可见的窗体显示出来。当把光标放置到“窗体布局”窗口的某个窗体上时，其形状变为一个四向箭头形。按下鼠标按钮，可以将窗体定位在希望它出现的地方。

当重新调整“窗体布局”窗口时，每个窗体都将相对于设计窗口的大小进行调整。窗口左上

角代表桌面坐标的(0,0)点。

2.4 控件的常用属性

尽管 VB 的控件种类繁多,每种控件又都有许多不同的属性,但是在这些众多的属性中,有相当一部分属性是大多数控件所共有的。因此,掌握这些共有属性对掌握控件的使用与设置会起到举一反三的效果。

(1) Caption

Caption 属性也称为标题属性,它决定了在控件上显示的内容。许多可视控件都有该属性。标题的最大长度为 1 024 个字节。

(2) Name

所有的控件在刚创建时都有一个默认的名称(如第 1 个命令按钮的名称为 Command1),为了便于编程与记忆,通常要对这个名称进行修改,即设置其 Name 属性。

与为变量命名相似,在为控件命名时,一般要遵守约定:控件前缀 + 自定义名。控件前缀表示了控件的类型,由于每个控件都有一个默认的名称,因此可从该默认名称中取出 3 个字符作为类型名;用户自定义名时最好有一定的意义,以便于记忆。名称的长度不超过 40 个字符。如"txtXsxm"表示学生姓名的文本框,"cmdExit"表示退出命令按钮。表 2-3 列出了部分控件的推荐前缀。

表 2-3 推荐使用的控件前缀

控件类型	中文含义	前缀	实例
CheckBox	复框	chk	chkReadOnly
ComboBox	组合框	cbo	cboEnglish
CommandButton	命令按钮	cmd	cmdExit
DriveListBox	驱动器列表框	drv	drvTarget
FileListBox	文件列表框	fil	filSource
Form	窗体	frm	frmEntry
Frame	框架	fra	fraLanguage
HorizontalScrollbar	水平滚动条	hsb	hsbVolume
ListBox	列表框	lst	lstPolicyCodes
Menu	菜单	mnu	mnuFileOpen
OptionButton	单选按钮	opt	optGender
PictureBox	图片框	pic	picVGA
Shape	形状	shp	shpCircle
TextBox	文本框	txt	txtLastName
Timer	定时器	tmr	tmrAlarm
VerticalScrollbar	垂直滚动条	vsb	vsbRate

对于上面没有列出的控件，应该用唯一的由两个或3个字符组成的前缀使它们标准化，以保持一致性。只有当需要澄清时，才使用多于3个字符的前缀。

前缀在一定程度上避免了名称冲突。另一方面，也不要使前缀太长以至于名称变得晦涩。

(3) Enabled

Enabled 属性决定了在运行时是否允许对象响应用户的操作。它的默认值为“True”，表示允许对象对用户的操作做出响应；若设置为“False”，则禁止对用户的操作做出响应，此时只要是可视的对象均呈灰色。几乎所有的对象都有 Enabled 属性。

(4) Visible

Visible 属性决定了对象在运行时是否可见，它也采用逻辑值，“True”代表可见，“False”代表不可见，但它不会影响其在设计时的可见性。只有可视对象才有 Visible 属性，像 Timer 那样不可视的对象没有 Visible 属性。

(5) Font

Font 属性决定了字符的格式，如字体、字号和字形等。尽管它是一个属性但又是一个对象，该对象包含了有关的字符格式属性，主要有以下几种。

- FontName：字体名称。
- FontSize：字号。
- FontBold：是否粗体。
- FontItalic：是否斜体。
- FontStrikethru：是否加删除线。
- FontUnderline：是否加下划线。

如果在属性窗口对该属性进行设置，只要通过字体对话框就能实现。而如果在代码中设置，则必须指出下级属性的名称，如：

```
txtXsxm. Font. FontName = "宋体"          '设置文本框的字体为宋体
```



字体名称必须与系统提供的完全一致。

(6) Height、Width、Top、Left

这4个属性给出了控件的大小与位置。Height 和 Width 属性指定控件的高度与宽度，Top 与 Left 属性指定控件在窗体上的位置。这些属性一般不需要进行精确设置，在实际设计时只要使用鼠标直接对控件操作即可以达到要求。在多个控件需要同时操作时，可以使用“格式”菜单下的“对齐”和“统一尺寸”两个菜单命令，详见后面的介绍。

(7) ForeColor、BackColor

这两个属性分别设置控件的前景色与背景色，其值是一个十六进制常数，通常使用调色板直接选取。

除了以上所介绍的属性外，还有一些属性也是大部分控件所共有的，但由于这些属性使用较少，因此不再赘述。

2.5 基本控件的使用

在 VB 应用程序的界面设计中，有3个基本的控件是最常用的，它们是标签、文本框和命令

按钮。这 3 个控件是构成程序界面的基础。本节将介绍这 3 个控件的使用方法，其他的控件在第 6 章介绍。

2.5.1　标签

标签(Label)用于显示静态的、不可修改的文本信息。标签控件常用做窗体中某些没Caption 属性控件的标识说明，如文本框、列表框和组合框等。另外，标签还可为窗体添加说明文字(如向用户提供帮助信息的说明文字)。

1. 控制标签中显示的文本

在标签控件中实际显示的文本是由 Caption 属性控制的，该属性可以在属性窗口中设置，在执行程序时用代码随时改变标签控件中所显示的文本，以适应不同的情况。

对标签中的文本可以通过 Font 属性和 ForeColor、BackColor 属性来改变其字体、前景色和景色。

利用 Alignment 属性，可以改变标签文本相对于标签的对齐方式：

- 0 - 左对齐。
- 1 - 右对齐。
- 2 - 居中对齐。

通常，标签在运行时只能看到标签的文本内容，但当将 BorderStyle 属性设置成 1(也可以在码中进行)，那么标签就有了一个很像“文本框”的边框。图 2-5 列出标签的几种不同显示方式。

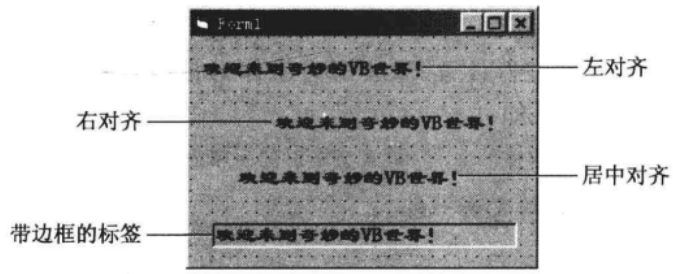


图 2-5　标签文本的显示

2. 标签大小自适应于文本

标签的大小由绘制时的大小确定，当然也可以通过 Height、Width 属性进行设置。当标签大小确定后，如果需要显示的文本内容超出了标签的范围，一般情况下多余的内容将不可见，那么如何使标签能自适应于所显示的内容呢？

标签控件提供了可以随时改变控件尺寸，以便帮助用户适应较长或较短标题的两种属性 AutoSize 和 WordWrap。

① AutoSize 属性决定了控件是否可以自动改变尺寸，以适应其内容的变化。如将该属性为 True，标签控件便会根据其内容改变水平方向的尺寸。

② WordWrap 属性可使“标签框”根据其内容自动进行垂直方向的改变，并保持其宽度不变。

如果要使 WordWrap 属性有效，AutoSize 属性也必须为 True。

2.5.2 文本框

文本框(TextBox)是 VB 中使用最为频繁的控件之一,它是一个文本编辑区域,程序运行时可以用来显示、输入和编辑文本。

文本框的属性很多,有些属性是前面已经介绍过的基本属性,如 Name、Height、Width、Enabled、Visible 与 Font 等,本节主要介绍直接影响文本框应用的属性。

1. 控制文本框中显示的文本

文本框最重要的属性当属 Text 属性,该属性中存放了文本框中显示的实际文本。Text 属性有 3 种使用方法。

- 设计时在属性窗口中设置文本框初始显示的内容,默认内容为“Text1”。

☞ 在属性窗口中输入文本时,不能换行,即不能按 Enter 键。

- 在运行时通过代码设置或由用户直接输入。

如 1.4.1 节中的“Text1. Text = "欢迎进入奇妙的 VB 世界""语句。

- 返回文本框中当前内容。

当用户对文本框中的内容进行了输入、编辑修改后,可以用该属性将当前内容返回,例如:

```
strText = Text1. Text
```

如果想在文本框中显示或输入多行文本,可以将 MultiLine 属性设为 True。这样当用户在文本框中输入时,就具备了一般字处理程序的自动换行功能,即当输入的文本超出文本框时会自动换行。同时按快捷键 Ctrl + Enter 可以插入一空行。

☞ 文本框最多可容纳 2 048 个字符,但如果 Multiline = True,则输入的字符长度可达 32 KB。

如果需要显示的内容较多,而文本框的空间是有限的,此时可利用 ScrollBars 属性给文本框加上滚动条。ScrollBars 属性有 4 种取值:

- 0 - 无滚动条。
- 1 - 加水平滚动条。
- 2 - 加垂直滚动条。
- 3 - 同时加水平与垂直滚动条。

显然,给文本框加上适当的滚动条并将 MultiLine 属性设为 True,此时的文本框与一般字处理程序的工作界面很相似。事实上,利用文本框控件可以方便地构成一个字处理程序的工作区,这也正是本章案例制作的基础。

❗ 如果加上了水平滚动条,文本框的自动换行功能就会消失,要想换行只能按 Enter 键。

除了以上叙述的两个属性外,还有两个属性是经常要用到的:Locked 属性与 PasswordChar 属性。Locked 属性指定文本框是否能被编辑,如果将该属性设置为 True,则文本框就相当于标签了,只能显示而不能被编辑。PasswordChar 属性主要用于在输入密码时将用户输入的字符用“*”屏蔽,但文本框 Text 属性的内容仍然为用户所输入的字符。

2. 对文本框的内容进行选择

利用文本框的 SelStart、SelLength 与 SelText 属性可以标识用户所选择的文本内容。

- SelStart:返回或设置所选择文本的起始点;如果没有文本被选中,则指出插入点的位置。

- SelLength：返回或设置所选择的字符数。
- SelText：返回或设置包含当前所选择文本的字符串；如果没有字符被选中，则为零长度字符串("")。

这3个属性在设计时都不能使用，仅在运行时可以使用。它们经常与剪贴板配合使用，以完成文本的剪切、复制和粘贴。

3. 控制文本框的事件与方法

文本框可以响应许多事件，其中最重要的是Change事件。当用户改变了文本框中的内容或通过代码改变了文本框的Text属性时，该事件就会被触发。显然，利用该事件可以跟踪文本框的内容变化以判断其内容是否有更改。

文本框本身是可以接收焦点的，只有当文本框具有焦点时，用户才能对其进行编辑或输入。文本框获得焦点时将触发GotFocus事件，一旦文本框失去焦点，将会触发LostFocus事件。利用该事件可以对文本框的内容更新进行确认或验证。

焦点的移动一般是由于用户的操作而引起的，但在代码中也可以控制焦点。文本框中有一个SetFocus方法能使文本框获得焦点。尤其当窗体上具有多个可以接收焦点的控件而又要在代码中控制焦点时，这个方法将非常有用，如：

```
Text1. SetFocus        '将焦点置于Text1文本框上
```

【例2-1】 按图2-6所示的界面布局设计一个窗体。

程序要求：txtText1为多行带垂直滚动条，用于程序运行时用户输入文本(文本字符为棕色、五号、楷体)。txtText2为一般文本框，只读，主要用于显示信息。当txtText1获得焦点时显示“光标在第一个文本框中”字样；当txtText1中的内容变化时显示“第一个文本框的内容已经改变!”字样。

设计操作步骤如下：

① 界面设计。新建一个窗体取名“frmExam_2_1”。

② 选中工具箱中的标签和文本框控件，分别创建1个标签和2个文本框，并进行适当的布局整理，如图2-6所示。

③ 在属性窗口中为控件设置属性。属性设置如表2-4所示。

表2-4　例2-1控件属性设置

控件名称	属性名称	属性设置值
上文本框	Name	txtText1
	Multiline	True
	ScrollBars	2-垂直滚动条
	Font	楷体、棕色、五号
下文本框	Name	txtText2
	Locked	True
命令按钮	Caption	确定
	Name	cmdExit
	Font	隶书，粗体，小四号

④ 在代码窗口中编写事件过程代码。

按照要求，当第一个文本框获得焦点后，在第二个文本框将显示相关信息，这只需在第一个文

本框的 GotFocus 事件中为第二个文本框的 Text 属性赋值即可。当第一个文本框的内容改变后也要在第二个文本框中显示相应的信息,这显然要在第一个文本框的 Change 事件中为第二个文本框赋值。当单击“确定”按钮时将关闭窗体,关闭窗体的命令为“Unload Me”。具体的程序如下:

```
Private Sub cmdExit_Click( )   '命令按钮的单击事件代码
    Unload Me                  '关闭窗体
End Sub
```

```
Private Sub txtText1_Change( )   'txtText1 的内容改变事件
    txtText2. Text = "第一个文本框中的内容已经改变!"      '为第二个文本框赋值
End Sub
```

```
Private Sub txtText1_GotFocus( )   'txtText1 的获得焦点事件
    txtText2. Text = "光标在第一个文本框中"                '为第二个文本框赋值
End Sub
```

⑤ 运行调试程序。观察运行结果,如图 2-7 所示。

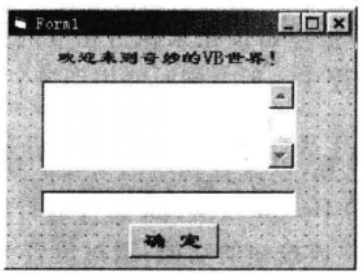


图 2-6 文本框示例布局图

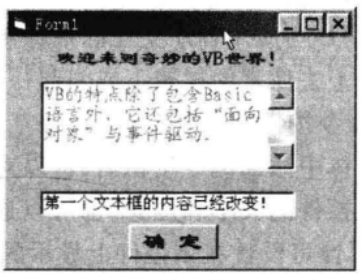


图 2-7 程序运行示例

将本例题以 frmExam_2_1 为名称并保存在 D:\Vbxx 文件夹中。

如果不加特殊说明,本书所有例题均存放在 D:\Vbxx 下,窗体文件的命名规则为:frmExam_章号_序号。

【例 2-2】 案例二初步设计。设计一个允许剪切、复制、粘贴和可进行简单编辑的便笺板。设计操作步骤如下:

① 界面设计。在工程 Vbxx 中新建一个窗体,从工具箱中选择文本框工具,在窗体上创建一个文本框作为便笺板的界面。

② 为控件设置属性。属性设置如表 2-5 所示。

表 2-5 案例二控件属性设置

控件名称	属性名称	属性设置值
窗体	Name	frmExam_2_2
	Caption	NoteEdit
文本框	Name	txtEdit
	Text	""
	Multiline	True
	Alignment	2 - Center
	ScrollBars	2 - Vertical

③ 编写事件过程代码。在 Windows 系列软件中，剪切、复制与粘贴都有通用的快捷键，VB 所提供的文本框也支持这种快捷键，因此本窗体基本不需要编写代码。

④ 运行调试程序。在运行该应用程序时，在文本框中输入正文，再选中一部分正文，用 Windows 的快捷键 Ctrl + X、Ctrl + C 和 Ctrl + V 进行剪切、复制和粘贴等操作，如图 2-8 所示。

将该窗体以 frmExam_2_2 为名保存在 Vbxx 工程中。

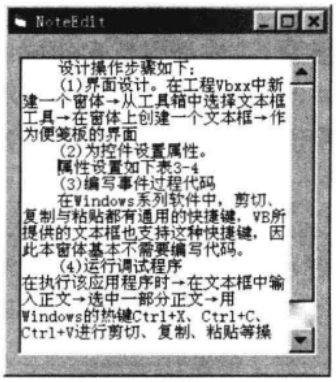


图 2-8　程序运行示例

2.5.3　命令按钮

命令按钮控件是应用程序界面设计中使用最为频繁的控件，几乎所有的应用程序中都要用到，它是用户与应用程序交互的最简单的方法。通过简单地单击按钮来执行相应的操作，同时该按钮看上去像真的被按下或弹起一样，非常直观。在程序中一般会同时使用到多个按钮。

1. 命令按钮的设置

向窗体中添加命令按钮后，可以对其基本属性进行设置，以符合程序的需要。如可以用鼠标调整该控件的大小和位置；也可以通过改变命令按钮的 Height 和 Width 属性进行调整；利用 Caption 属性可设置在该控件上显示的文字，以便告诉用户该按钮的功能；通过设置 Font 属性，可以改变显示文本的字体和大小。

(1) 设置按钮的访问键

在一般应用软件的对话框的命令按钮上，都设置有访问键(带下划线的字母)以便用键盘进行快速操作。为命令按钮设置访问键的方法是：在设置按钮的 Caption 属性时，在需要作为访问键的字母前加上“&”，则在程序运行时，该字母即会带上下划线而成为访问键。当用户按下 Alt + 访问键字母时，即相当于用鼠标单击该按钮。比如在图 2-9 中，两个按钮的 Caption 属性分别设置为“(O)确定”和“(C)取消”。

(2) 默认按钮与取消按钮的设置

默认按钮是指按 Enter 键时相当于单击该按钮，而取消按钮是指按 Esc 键时相当于单击该按钮。在窗体上只能有一个默认按钮与一个取消按钮。

要将某一个按钮设置为默认按钮，只要将该按钮的 Default 属性设为 True；要将某个按钮设为取消按钮，只要将该按钮的 Cancel 属性设为 True。图 2-10 中将“确定”按钮设为默认按钮，将“取消”按钮设为取消按钮。

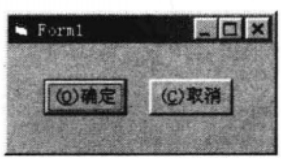


图 2-9　按钮的访问键

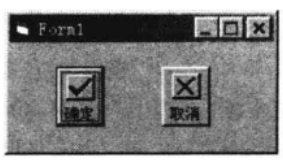


图 2-10　加载图片后的按钮

(3) 增强按钮的视觉效果

尽管可以通过命令按钮的 Font 属性来对按钮上的文字进行一些设置,但它毕竟还是文字。文字的表现力当然不如图片。要为按钮加上图片,请按以下步骤:

① 将 Style 属性设置为 1 - Graphical。

② 设置按钮的 Picture 属性,单击该属性右侧的"…"按钮,屏幕上将出现"加载图片"对话框。

③ 选择合适的图片文件并打开,该图片将显示在按钮的文字上方。

增加图片后按钮效果如图 2-10 所示。

☞ 在 VB 6.0 的安装文件夹下有一个 Common 文件夹,其中的 Graphics 文件夹中有大量的图片文件可供选用。当然,也可以自己制作图片文件。

☞ 如果不想在按钮上显示文字,只需将按钮的 Caption 属性清空。

(4) 为按钮增加动态提示

动态提示是指程序运行时,当用户用鼠标指向按钮时,在按钮右下方会显示指定的提示文字。这是利用控件的 ToolTipText 属性实现的。文本框按钮等许多控件都有 ToolTipText 属性。

2. 命令按钮的编程

命令按钮的事件中最主要的是 Click 事件,它是用户单击命令按钮时触发的。一旦触发 Click 事件,程序将会运行该事件过程中的代码。因此对命令按钮的编程主要是编写单击事件过程代码。

实际上除了鼠标的单击能够触发 Click 事件以外,以下动作同样会引发该事件:

- 按 Tab 键,把焦点移到相应的按钮上,再按 Space 键或 Enter 键。
- 按命令按钮的访问键(Alt + 带下划线的字母)。
- 在代码中设命令按钮的属性 Value 为 True。如:

```
cmdExit. Value = True
```

- 在代码中显式调用命令按钮的 Click 事件。如:

```
Command1_Click
```

- 若该命令按钮是窗体默认的命令按钮,即使将焦点移到其他不是命令按钮的控件上,按下 Enter 键也会选中该按钮而触发 Click 事件。
- 如果该命令按钮是窗体默认的取消按钮,即使把焦点移到其他控件上,也能通过按 Esc 键选中该按钮而触发 Click 事件。

2.5.4 案例二的进阶设计

例 2-2 已经对案例二进行了初步设计,现在窗体上再增加 4 个按钮:"剪切"、"复制"、"粘贴"和"退出",为每个按钮加载图片,并给出动态提示,以完成该案例的设计。

(1) 界面设计

打开 frmExam_2_2 窗体,在窗体上增加 4 个按钮,使用调整位置与大小的方法对 4 个按钮进行适当的调整。将 4 个按钮的 Caption 属性均设为空,Style 属性均设为 1 - Graphical,其他属性的设置如表 2-6 所示。

表 2-6　命令按钮的属性设置

控件名称	属性名称	设置值
命令按钮	Name	cmdCut
	Picture	Common\Graphics\Bitmaps\TlBr_W95\Cut. bmp
	ToolTipText	剪切
命令按钮	Name	cmdCopy
	Picture	Common\Graphics\Bitmaps\TlBr_W95\Copy. bmp
	ToolTipText	复制
命令按钮	Name	cmdPaste
	Picture	Common\Graphics\Bitmaps\TlBr_W95\Paste. bmp
	ToolTipText	粘贴
命令按钮	Name	cmdExit
	Picture	Common\Graphics\Bitmaps\TlBr_W95\Close. bmp
	ToolTipText	退出

(2) 编写按钮的事件过程代码

在例 2-2 中,剪切、复制和粘贴使用了 Windows 默认的快捷键的方法,实际上还可以使用编程的方法来实现,有两种方法。

- 方法一:使用发送按键消息。

在 VB 中有一个发送按键消息的语句 SendKeys,该语句的功能是将按键的消息发送到活动窗口,就如同直接在键盘上输入一样。其语法格式为:

```
SendKeys string
```

string 为按键消息的字符表达式,对能显示的单个字符键而言,直接用字符键本身的键名字符,对无显示的特殊键则需将键名字符放在"{}"中,而对 Ctrl、Alt、Shift 这 3 个特殊键则分别用^、%、+来表示。如 Ctrl + C 表示为"^C"。

这样,只要在剪切、复制和粘贴按钮的事件过程代码中直接向文本框发送 3 个快捷键的消息,即可实现以上功能要求。

具体代码如下:

```
Private Sub cmdCopy_Click()          '复制按钮的事件过程
    txtEdit. SetFocus                '将焦点置于文本框
    SendKeys "^C"                    '发送"Ctrl + C"快捷键
End Sub
```

```
Private Sub cmdCut_Click()           '剪切按钮的事件过程
    txtEdit. SetFocus
    SendKeys "^X"                    '发送"Ctrl + X"快捷键
End Sub
```

```
Private Sub cmdExit_Click()          '退出按钮事件过程
    Unload Me                        '关闭窗体
```

```
End Sub
```

```
Private Sub cmdPaste_Click( )          '粘贴按钮的事件过程
    txtEdit. SetFocus
    SendKeys "^V"                      '发送"Ctrl + V"快捷键
End Sub
```

- 方法二:利用文本框属性编程。

文本框具有 SelStart、SelLength 和 SelText 属性,专门用来标识用户所选择的字符,因此利用这 3 个属性也可以实现上述功能。具体代码如下:

```
Dim strTemp As String                  '定义存放选中字符串的变量

Private Sub cmdCopy_Click( )
    strTemp = txtEdit. SelText         '将选中的文本送临时变量
End Sub
```

```
Private Sub cmdCut_Click( )
    strTemp = txtEdit. SelText
    txtEdit. SelText = ""              '将选中的文本清空
End Sub
```

```
Private Sub cmdExit_Click( )
    Unload Me
End Sub
```

```
Private Sub cmdPaste_Click( )          '重新生成文本框中的文本
    txtEdit. Text = Left(txtEdit. Text, txtEdit. SelStart) _
  + strTemp + Mid(txtEdit. Text, txtEdit. SelStart + 1)
End Sub
```

(3) 程序运行

将窗体保存为 frmExam_2_3,然后运行,其结果如图 2-1 所示。

习 题 2

1. 什么是焦点? 哪些对象可以获得焦点? 怎样才能获得焦点?
2. 在 VB 中有哪几类控件? 分别有何作用?
3. 如何调整控件每次移动的距离?
4. VB 控件的常用属性有哪些? 它们有什么作用?
5. 如何设置控件的属性? 何为静态设置? 何为动态设置? 分别用于什么情况?
6. 同时对一组同一类控件的某个属性进行同样的设置应如何操作?
7. 编一个程序,运行时在屏幕上显示一个窗体。单击窗体时显示一幅图,右击时清除该图。
8. 如何将一个窗体设为启动窗体?
9. 什么是访问键? 如何为命令按钮设置访问键? 怎样为文本框设置访问键?

10. 默认按钮与取消按钮的含义是什么？如何设置？

11. 设计一个窗体，其包含一个命令按钮“退出”和一个文本框。

要求：运行开始时命令按钮不能用，只有当文本框中输入了“请输入信息”的文本后，才能单击命令按钮退出。

12. 设计一个窗体，要求每单击一次按钮或单击图标本身，即可变换出不同颜色的交通信号灯图标，如图 2－11 所示。

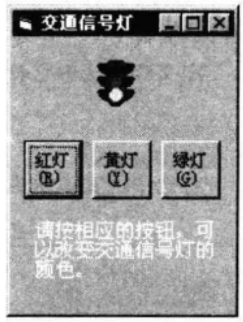


图 2－11　程序运行示例

第3章　VB程序设计语言基础

学习目标

掌握VB中的数据类型、常量与变量的使用，能熟练地使用数组、常用标准函数与人机交互函数。熟悉程序的3种基本控制结构，能根据应用需要进行一般的程序设计。学会VB中的代码编制方法，为以后的学习打下良好的基础。

从前两章的学习可以看出，一个VB程序由两个部分构成：一部分是程序的应用界面，另一部分是程序代码。程序的应用界面正是通过程序代码将界面中的各个对象有机地结合起来，从而实现程序所需要的功能。与其他任何程序设计语言一样，VB也规定了可编程的数据类型、基本语句和函数与过程等。

对于已经学习过其他编程语言的读者来说，本章的许多内容是不陌生的，因此学习时只要快速浏览一下即可，而对于没有编程语言基础的读者来说，本章则应是学习的重点。

3.1　模块与代码窗口

3.1.1　模块

在VB中，所有的程序代码均放置在不同的模块中。依据不同的应用场合，模块有3种类型，即窗体模块、标准模块和类模块。

1. 窗体模块

应用程序是由若干个窗体构成的，而每一个窗体都对应一个窗体模块。在窗体模块中不仅包含着对窗体及其控件属性的设置以及相关的说明，而且包含着更重要的用于处理发生在窗体中各个对象事件的过程。每个窗体模块都存放在一个扩展名为.frm的文件中。默认情况下，应用程序只有一个窗体，但可以通过“添加窗体”按钮添加新的窗体。

2. 标准模块

如果程序代码全部放在窗体模块中，那么这些程序代码只能被该窗体本身所调用。而对于一个完整的应用程序来说，一般不可能只有一个窗体。当应用程序包含有多个窗体时，就会出现有多个窗体都要执行一段或几段公共代码。显然，将这些代码在每个所需要的窗体中均重写一遍是极不科学的。为了解决这个问题，VB提供了一种称为“标准模块”的独立模块。

标准模块的扩展名为.bas，其中可以包含变量、常量、外部过程及声明等。在标准模块中出现的过程代码不仅能被应用程序的其他模块访问，而且还可以供其他应用程序使用。

如要向一个工程中添加标准模块，只要选择〖工程〗→〖添加模块〗菜单命令或单击“添加”按钮中的“添加模块”按钮即可。

3. 类模块

类的编程与应用是所有面向对象应用程序设计的基础，VB 也不例外。类是一个模板，对是由它创建的。VB 的应用程序设计中，一方面可以使用 VB 中现成的对象，另一方面还可以过类模块来创建新的对象，这些新的对象中包含有自己的各种属性与方法，它可以在应用程中使用。类模块与标准模块最大的不同在于标准模块只包含代码，而类模块既包含代码又包数据，它可视为没有物理表示的控件。

类模块保存在扩展名为 .cls 的文件中。有关类模块的详细内容，感兴趣的读者可以参阅他的相关资料，本书不再涉及。

3.1.2 过程

如前所述，VB 程序代码均放置在相应的模块中，那么它们在模块中又是如何组织的呢？来，VB 是将所有的程序都组织在若干个“过程”中的。

从前几章的示例中可以看到，所谓“过程”实际上是具有特定书写格式并且包含一组具有定功能的代码集合。根据执行的方式，可把过程分为事件过程与通用过程两大类。事件过程 VB 中最基本的过程，由于 VB 采用的是事件驱动的编程机制，因此由用户或系统触发了某个件后，总要执行相应的代码来完成相应的功能，这就是事件过程。事件过程总存放在某一个体模块中。通用过程与事件过程不同，它不属于任何一个窗体，但可以被多个窗体所共享，这可被共享的代码构成的过程即称为通用过程。有关过程的详细描述请参见第 6 章。

3.1.3 代码窗口及其使用

代码是由语句、常量、变量和声明等部分组成。代码窗口是用于编写程序代码的场所，V应用程序的每个模块都有一个代码窗口与之对应。因此，要想为某个模块编写代码，必须首打开该模块对应的代码窗口。

使用下列操作方法之一，即可打开代码窗口：

- 双击要编写代码的窗体或控件。
- 从“工程资源管理器”窗口选定窗体或模块的名称，右击，然后从快捷菜单中选择“查代码”命令。
- 用鼠标单击窗体或控件，在菜单栏上选择“视图”→“代码窗口”命令。

图 3-1 即为打开的代码窗口，图中的标题栏显示的是工程名称与窗体名称，其下有两个拉列表框，左边的下拉列表框中包含了该窗体中的所有对象及一个“通用”对象。通过该下拉表框可以选择要编写代码的对象。通用对象中主要放置该窗体模块所共享的数据定义、常量明、类型声明以及通用的过程等。右边的下拉列表框包含了所选中对象的各个事件，通过选不同的事件即可在相应的过程框架中编写该事件的过程代码了。

代码窗口的主工作区即为编写代码的场所，要注意的是，当选择了某个对象中的某个事后，代码窗口会自动给出相应的过程框架，所编写的代码必须包含在过程框架之中。每个事过程都含有一个独立的代码段。

如要打开通用模块，则必须先添加模块，添加后通用模块的代码窗口会自动打开，如图 3-2 所

要修改程序代码的内容，当然也要先打开相应模块的代码窗口。方法是：在“工程资源管

”的窗口中右击需要修改代码的项目，在快捷菜单中选择“查看代码”命令。通过在代码窗口选择不同的对象和不同的事件，可以查看、修改不同的过程。



图 3-1 窗体模块代码窗口

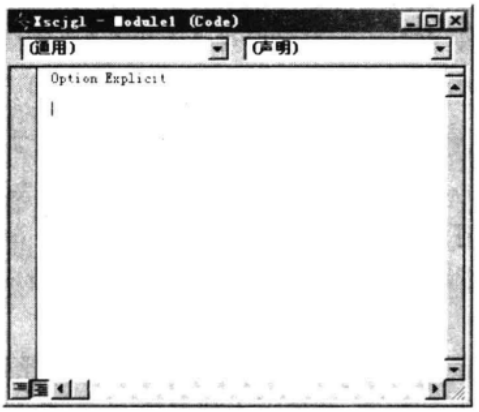


图 3-2 通用模块代码窗口

无论是哪种代码窗口，都有两种查看方式：一是每次仅查看一个过程，二是查看代码模块中的所有过程。两种查看方式的切换是通过代码窗口下水平滚动条左侧的“过程查看”与“全模块查看”按钮来实现的。

VB 的代码窗口具有高度智能化的编辑特性，主要体现在以下几个方面。

（1）自动语法检查

当一行代码输入完成并将光标移动到其他行时，VB 将对该行代码进行语法检查。如没有语法错误将自动进行格式编排；如存在语法错误，屏幕会弹出错误消息框，提示错误的类型同时该行变为红色。

（2）自动格式编排

VB 会对输入完成且没有错误的代码行进行自动的格式编排。对 VB 中的关键字、注释字符、非关键字等语言要素采用不同的颜色以示区别，对 VB 中的所有关键字的第一个字母改为大写，其后变为小写，对各个语言要素之间加上适当的空格。

（3）自动编码

自动编码功能体现在两个方面。一是“自动列出成员”功能，即当在代码窗口输入某个对象名称时，屏幕会自动列出相应对象的所有属性和方法。输入属性或方法名称的前几个字母便可从列表中选择该名称，按下 Tab 键即可完成属性名称的输入。这样，一方面大大加快了代码的输入速度，另一方面减少对属性名称的记忆量。自动编码的另一个功能是“自动显示快速信息”功能，即在输入合法的语句或函数名称后，屏幕会显示该命令或函数相应的语法，这样就可以根据提示完成相关参数的输入，减少了不必要的错误。

以上这些自动功能均可通过选择〖工具〗→〖选项〗→〖编辑器/编辑器格式〗菜单命令进行设置。

3.1.4 编码规则与注释

1. 编码规则

VB 的程序代码与其他任何程序设计语言的代码一样，都要遵守一定的书写规则，否则所书写的

代码将不能被正确识别，从而产生编译或运行错误。在 VB 中编写代码时应遵守以下几方面的约定：

① 代码输入时可以不区分大小写，由系统自动进行转换。

② 在同一行上可以书写多条语句，但语句间必须用“:”分隔。

③ 每行的字符最多可达 1 024 个。为便于阅读和打印，可以将长语句分成多行，但必须在行尾加上续行符（空格再加上下划线“_”），而且续行符后不能再加注释。

☞ 原则上，不提倡一行写多条语句，续行符应该加在运算符的前面或后面。

④ 对数字除了可以用默认的十进制表示外，还可以用其他进制，但必须加上前缀说明：十六进制用 &H（如 &H9A），八进制用 &O（如 &O76）。

⑤ 对过程、循环和分支的程序段一般应进行缩排处理，从而便于阅读与调试。

2. 注释

给某些关键性的代码加上必要的注释有利于程序的阅读、维护与调试。VB 中的注释是以“'”符号开始的，有如下 3 种加法：

- 在代码行的尾部加，但不能出现在续行符后。
- 注释单独为一行或几行，每行前都必须有注释符。
- 利用编辑工具栏对某一个程序段整个加上注释。方法是先选中需要注释的代码段，单击编辑工具栏中的“块注释”按钮，如要取消注释，只要再单击“取消块注释”按钮即可。这种注释特别有利于程序的调试。

3.2　VB 中的数据与运算

数据与数据运算是任何一种程序设计语言编程的基础。VB 提供了丰富的数据类型和运算方式以适应不同的场合。

3.2.1　数据类型

在 VB 中共有 11 种系统数据类型与一种自定义类型，各种类型数据的基本情况如表 3-1 所示。

表 3-1　VB 的数据类型

数据类型	关键字	类型符	前缀	占字节数	应　用
字节型	Byte	无	byt	1	存储二进制数据，范围较小，不能存放负数
逻辑型	Boolean	无	bln	2	只有 True 和 False 两种取值，用于逻辑判断
整型	Integer	%	int	2	保存整数，取值范围小，用于不带小数的数据
长整型	Long	&	lng	4	取值范围大于整型，其余同上
单精度型	Single	!	sng	4	保存浮点实数，运算可能会产生误差
双精度型	Double	#	dbl	8	取值范围大于单精度型，其余同上
货币型	Currency	@	cur	8	保存定点实数，主要用于货币计算
日期型	Date	无	dtm	8	保存日期与时间数据
字符型	String	$	str	可变	保存字符串数据，分为定长与变长两种
对象型	Object	无	obj	4	存储 32 位的地址数据，该地址可引用应用程序中的对象

续表

数据类型	关键字	类型符	前缀	占字节数	应　用
变体型	Variant	无	vnt	可变	可以保存所有系统定义的数据类型;主要是用作未定义变量的默认数据类型,其数据的处理完全取决于上下文

说明:

① 用于处理数值的数据类型最多,共有 6 种:字节型、整型、长整型、单精度型、双精度型和货币型。各种类型的取值范围、运算速度、运算精度都不同。从取值范围看,双精度型与货币型最大,字节型最小;从运算速度看,整型最快;从运算精度看,整型高而实数类型的较低。因此,使用时要根据不同的情况选择合适的数据类型。例如,要保存年龄、工龄等数据可选用整型,而要保存成绩、工资等数据可选用单精度型。

② 对字符数据类型 String 而言,一般情况下其中保存的字符串长度是可变的,当然也可以使用定长的字符串。例如:

```
Dim Str1 As String          '声明变长字符串
Dim Str2 As String * 20     '声明长度为 20 的定长字符串
```

③ 日期型数据类型表示的有效范围为:日期从公元 100 年 1 月 1 日至 9999 年 12 月 31 日,时间从 00:00:00 到 23:59:59。

④ VB 中最为智能的是变体型 Variant,它通常被称为万能型。当某一个变量指定为 Variant 型时,不必在数据类型之间进行转换,VB 会自动完成各种必要的转换。当然系统的处理效率会有所降低。

⑤ 各种数据类型之间是可以相互进行转换的。下一节将介绍这些转换函数。

3.2.2 常量与变量

任何一个应用程序都要对数据进行各种各样的处理。有些数据在程序的运行过程中是始终保持不变的,这些始终不变的数据称为“常量”;而有些数据在程序中其取值是变化的,这些数据称为“变量”。

1. 常量

常量也具有不同的数据类型,不同数据类型的常量表示形式是不同的。

数值型常量由正负号、小数点和数字构成。对于十进制数的表示与日常使用的形式是基本相同的,如 123、-45.65 等。但对于非十进制数应该加上表示进制的前缀,如 &H2A5 表示该数是十六进制数 2A5。

字符型常量必须用引号("")括起来,如"Visual Basic"、"数学"等。

日期型常量的完整表示形式为“#月/日/年 时:分:秒 AM 或 PM#”,如#07/21/2001#、#07/13/2001 10:11:45 PM#等都是合法的日期。

VB 不接受包含汉字的日期格式数据,如“#2001 年 10 月 29 日#”这样格式的日期。

为了提高代码的可阅读性与可维护性,VB 中可以将常量用固定的符号来表示,这就是符号常量。其语法如下。

格式:[Public | Private] Const constname [As type] = expression

功能：声明一个常量。

说明：constname 为常量的名称。Public | Private 表示所声明常量的作用范围，Public 为全局常量，Private 为模块级常量。expression 为合法的 VB 表达式。

例如：

```
Const conPi = 3.14159
Public Const conUser As String = "Admin"
```

VB 中有许多内部或系统常量，这类常量的名称通常用前缀来表示定义常量的对象库名，如来自 VB 或 VBA 的常量以前缀 vb 开头，来自 Excel 的常量以 xl 开始。

2. 变量

变量常用来存放程序运行时的临时数据，这些数据的值在设计时是未知的，运行中随时都可能发生变化。

(1) 变量的命名

为了便于在程序中加以区分，每一个变量都必须有一个唯一的名字，给变量命名时要遵守以下规定：

① 必须以字母开头，由字母、数字和下划线构成，长度不超过 255 个字符，大小写不区分。

② 最后一个字符可以是表 3-1 中相应的类型符。

③ 不能与 VB 的保留字重名，如 For、Do、End、Sub、Function 和 While 等。

④ 为了增加程序的可读性，变量的命名应含义清楚，同时尽量采用前缀来表示变量的数据类型。前缀可参照表 3-1 中的说明。例如，strName、dblJe、blnHf 和 dtmCsrq 等分别表示的是字符型的姓名、双精度型的金额、逻辑型的婚否和日期型的出生日期等变量。

(2) 变量的声明

每个变量在使用时都要占用一定的内存空间，为便于系统分配存储空间，变量在使用时都要声明其类型。声明变量的方法有 3 种。

① 隐式声明。指程序中第一次出现某个变量时，系统自动创建该变量。隐式声明的变量类型为 Variant 型。例如：

```
Private Sub Form_Load()
    TestToday = Now()
    Debug.Print TestToday                '将结果输出到立即窗口
End Sub
```

该代码段的第一行中，VB 用 TestToday 自动创建了一个变量，第二行就可以直接使用了。尽管隐式声明使用非常方便，几乎不加限制，但如果将变量名写错，将会产生一个难以查找的错误。例如，将上述程序的第二行写成“Debug.Print TesToday”，看起来似乎没问题。但由于将 TestToday 写错了，VB 显然无法显示当前的日期与时间。这是因为 VB 分辨不出究竟是隐式声明一个新变量，还是仅仅将现有变量名写错了，于是只好再创建一个新的变量。所以，为避免这种错误的发生，一般要求对使用的变量都要先进行声明，即采用显式声明。

② 显式声明。显式声明就是在变量使用前先使用声明语句(如 Dim)来声明变量。一般来说，变量的声明总是放在模块的“声明”部分。以 Dim 为例：

格式：Dim variablename [As type]

如：

```
Dim strName As String
Dim blnHf As Boolean,dtmCsrq As Date
```

为了防止不声明就使用变量,可以在每个代码模块前写上强制进行显式声明的语句:

```
Option Explicit
```

这样,系统一旦遇到没有声明的变量时,会发出错误警告。另外还有一种办法,即选择〖工具〗→〖选项〗菜单命令,在弹出的对话框中打开"编辑器"选项卡,选中"要求变量声明"复选框。这样,以后在工程中的任何新模块的开始,都会自动加上"Option Explicit"语句。

这种办法只对新的模块起作用,对已有的模块还必须手工加入。

③ 类型声明符。在表3-1中有6种数据类型具有类型符,它主要是用做隐式声明时来说明变量的类型。如strName $ = "张三"表示strName是一个字符型变量,其初值为"张三"。

3.2.3 运算符与表达式

1. 运算符

VB的运算符有4类:算术运算符、字符串运算符、关系运算符和逻辑运算符。

(1) 算术运算符

与一般的程序设计语言类似,VB包含加(+)、减(-)、乘(*)、除(/)、整除(\)、取余(Mod)、乘方(^)和负(-)等运算符。各种运算符的含义比较明显,不再赘述。

(2) 字符串运算符

字符串运算也称为连接运算,有"&"与"+"两种运算符。两种运算符都是将运算符两边的字符串拼接起来。"&"与"+"的主要区别在于:"+"两边都应为字符型数据,而"&"则无此限制,当两边的类型不同时,"&"运算符首先将操作数转换成字符型再进行连接。

使用"&"运算符时,如操作数为变量,在变量与"&"符号之间必须加一空格,以防与类型符冲突。

(3) 关系运算符

关系运算用来比较两个表达式的大小,有大于(>)、大于等于(>=)、等于(=)、小于(<)、小于等于(<=)和不等于(<>)6种运算符。关系运算的结果为逻辑值,它不仅可以用于数值型数据的比较,而且还可以用于字符型数据与日期型数据的比较。字符数据以其ASCII值从左到右进行比较,直到比较出结果,而日期数据以其日期值进行比较。

(4) 逻辑运算符

逻辑运算也称布尔运算,专门用于逻辑值的运算,运算的结果也为逻辑值。逻辑运算符的含义如表3-2所示。

表3-2 VB的逻辑运算符

运算符	含义	优先级	举例	结果	说明
Not	非	1	Not(3>2)	False	对操作数取非
And	与	2	(3>2) And ("a">"b")	False	两个操作数都为真时,结果才为真
Or	或	3	(3<2) Or ("a">"b")	False	有一个操作数为真,结果即为真

续表

运算符	含义	优先级	举　例	结果	说　明
Xor	异或	4	(3>2) Xor ("a">"b")	True	两个操作数不等时才为真
Eqv	等价	5	(3>2) Eqv ("a">"b")	False	两个操作数相等时才为真
Imp	蕴涵	6	(3>2) Imp ("a">"b")	False	第一操作数为真,第二操作数为假,结果才为假,其余均为真

在表达式中,当运算符不止一种时,优先处理算术运算符,接着处理比较运算符,然后再处理逻辑运算符。所有比较运算符的优先顺序都相同,也就是说,要按它们出现的顺序从左到右进行处理。算术运算符按一般的算术习惯进行,逻辑运算符则必须按表3-2中所说明的优先级进行处理。当然也可以用括号来改变优先顺序。

2. 表达式

由变量、常量、函数和运算符按一定的规则构成的一个字符序列就是表达式,每个表达式都有一个运算结果,运算结果的类型取决于参与运算的数据类型和运算符。以下是几个典型的表达式实例。

```
a * b/(c + d)                                          '算术运算,结果为数值型
strB$ < >"Visual Basic"                                '关系运算,结果为逻辑型
intAge < 19 And sinTotal > 285 And (kccj1 = 100 Or kccj2 = 100)    '综合运算,结果为逻辑型
```

3.2.4　数组

前面所介绍的各种数据类型都属于简单类型,通过一个命名的变量只能存取一个数据。然而,在实际应用中经常要处理一批同一性质的数据,如100个学生的某门课程的成绩。此时,如果再用普通的变量来处理就显得力不从心了,为此VB引入了数组的概念。VB中将具有相同名字、下标值不同的一组变量称为数组变量,简称数组。

数组中的每个变量称为数组元素,数组元素的表示方法为:数组名(s1,s2,…)。其中s1、s2等称为下标,只有一个下标的数组称为一维数组,有两个下标的数组称为二维数组,等等。下标值只能是整型数。VB中的数组有两种:一种是静态数组,另一种是动态数组。数组必须先声明后使用。数组的声明既可以在模块的声明段声明,也可以在过程中声明。

1. 静态数组

所谓静态数组是指数组的维数与下标的范围在声明时就已经确定了。声明静态数组使用的语句与声明变量的语句类似。

【例3-1】 静态数组的声明。

```
Dim sngCj(45) As Single              '声明用于存放学生成绩、具有46个元素的一维数组
Dim intA(3,4) As Integer             '声明一个二维的整型数组
Dim BirthDay(1 To 10) As Date        'BirthDay是一个下标从1到10的Date型数组
```

① 数组名中的下标值是指该数组下标的上限,下限默认从0开始,如不想从0开始,可以采用第三种声明方式。

② 在数组声明中出现的下标代表的是数组整体,而在其他地方出现的下标则代表是数组中的一个元素。

2. 动态数组

动态数组是指在声明数组时不指明下标的大小(省略括号中的下标),当需要使用时,再用 ReDim 语句重新定义其大小。建立动态数组的步骤如下。

① 在模块或过程中声明动态数组,例如:

```
Dim intRs( ) As Integer
```

② 在需要使用该数组的过程中分配实际的元素个数。

```
ReDim intRs(x + 1)   'x 是一个整型变量
```



可以多次使用 ReDim 语句来改变数组的大小,但每次使用都会使原数组的内容丢失,除非使用 Preserve 关键字而且不能用该语句来改变数组原有的数据类型。

Option Base 语句的语法格式为:

```
Option Base {0|1}
```

需要说明的是,默认的数组下标下限为 0,不需要使用该语句;添加上述语句后,可将数组下标的下限更改为 1。另外,该语句必须放置在模块的所有过程和所有数组声明之前,而且一个模块中只能出现一次。

在下面这个例子中,使用 Option Base 语句来修改数组下标的默认值。其中,LBound 函数返回数组指定维的下标下限值。

```
Option base 1    ' Set default array subscripts to 1.

Dim Lower
Dim MyArray(20), TwoDArray(3, 4)          '声明数组变量
Dim ZeroArray(0 To 5)                     '修改默认的数组下标下限
'使用 LBound 函数测试数组下标的下限值
Lower = LBound(MyArray)                   '返回 1
Lower = LBound(TwoDArray, 2)              '返回 1
Lower = LBound(ZeroArray)                 '返回 0
```

3. 数组的使用

数组在声明后即可使用,数组的使用主要涉及两方面的问题:一是数组元素的赋值,二是数组元素的访问。实际上,这是一个如何访问数组中元素的问题。要访问数组中的元素应该使用如下的格式:数组名(下标值)。

数组的赋值类似于变量的赋值,由于数组的元素较多,因此对数组的赋值通常都要与循环程序相结合来完成赋值。数组元素的引用也必须通过下标进行。

【例 3-2】 数组的使用。

```
Private Sub Form_Click( )
  Dim intA(10) As Integer
  Dim i As Integer
  For i = 0 To 10
     intA(i) = 1   '对每个数组元素赋初值 1
     Print intA(i)   '输出数组元素的值,也即数组元素的引用,将结果输出到当前窗体
  Next i
End Sub
```

在 Vbxx 工程中新建一个名为 frmExam_3_2 的窗体，在窗体的 Click 事件中输入以上代码。

在引用数组元素时，下标不能超出其上下界，否则将产生一个运行错误。

在 VB 中，除了静态数组与动态数组外，还有一种特殊的数组，即控件数组。有关控件数组的内容详见第 5 章。

3.3　标 准 函 数

函数是指能实现特定功能或完成某种运算的程序段，如求某个数的平方根、取得系统的时间和日期等。VB 中的函数分为内部函数和用户自定义函数（见 4.2.1 节的 Function 过程）。

内部函数（也称标准函数）可供用户在编程时直接调用。内部函数按其功能分为数学函数、字符串函数、转换函数、日期函数和交互函数等。每个函数都有一个系统规定的函数名，函数一般总是出现在表达式中，函数调用的格式如下：

<函数名>(p1,p2,…)

其中，p1、p2 等为调用函数的自变量序列，也称为函数的参数，其个数、类型及排列次序都必须与函数规定的参数相同。

3.3.1　常用内部函数

表 3-3 列举了 VB 中典型内部函数的应用。在以后的叙述中，用 N、C、D 分别表示数值类型、字符类型、日期类型参数，凡函数名后带有“$”者表示该函数返回字符类型。有关函数的种类及具体使用方法请参见 VB 的联机帮助。

表 3-3　几种典型的 VB 内部函数应用举例

函数名	功能	类型	举例	结果
Abs(N)	取绝对值	算术函数	Abs(-12.34)	12.34
Int(N)	取整		Int(-99.2)	-100
Fix(N)	取整		Fix(-99.2)	-99
Log(N)	返回参数的自然对数值		Rnd	0~1 之间的任意数
Exp(N)	返回 e 的 N 次幂		Sqr(16)	4
Rnd([N])	产生随机数			
Sqr(N)	平方根			
Asc(C)	转换成 ASCII 值	转换函数	Asc("A")	65
Chr[$](N)	ASCII 值转换为字符		Chr(65)	"A"
Str[$](N)	数值转换为字符		Str(45.90)	"45.9"
Val(C)	字符转换为数值		Val("23.4a")	23.4
Len(C)	字符串长度	字符串函数	Len("Visual Basic")	12
Left[$](C,n)	取字符串左边 n 个字符		Left("Visual Basic",6)	"Visual"
Right[$](C,n)	取字符串右边 n 个字符		Rigth("Visual Basic",5)	"Basic"
Ltrim[$](C)	去掉字符串左边的空格		Ltrim(" Basic")	"Basic"
Rtrim[$](C)	去掉字符串右边的空格		Rtrim("Visual ")	"Visual"
Trim[$](C)	去掉字符串两边的空格		Trim(" Visual ")	"Visual"
Mid[$](C,n1,n2)	从 n1 位开始取 n2 个字符		Mid("Visual Basic",9,2)	"as"

续表

函数名	功能	类型	举例	结果
	在字符串 C1 中查找 C2			
InStr(C1,C2)	将 C 转换成大写字母	字符串函数	InStr("Visual","a")	5
Ucase[$](C)	将 C 转换成小写字母		Ucase("Visual")	VISUAL
Lcase[$](C)			Lcase("Visual")	visual
Date[$]	取系统日期	日期时间函数	Date	当前日期
Time[$]	取系统时间		Time	当前时间
Now[$]	取系统日期和时间		Now	当前日期和时间
Day(C\|N)	取日期值		Day("07/13/2001")	13
Month(C\|N)	取月份值		Month("07/13/2001")	7
Year(C\|N)	取年份值		Year("07/13/2001")	2001

说明:加括号的 $ 可以省略。

3.3.2 字符处理

VB 的字符处理非常灵活,为了配合字符串的使用,VB 还提供了大量丰富的字符串函数。

1. 字符处理

字符必须编码后才能被计算机处理。早期的计算机使用 7 位的 ASCII 编码,为了处理汉字,又设计了用于简体中文的 GB 2312 和用于繁体中文的 BIG 5 编码。这样,一个文本字符在计算机中只占一个字节;而一个汉字在计算机中却要占两个字节。

GB 2312 支持的汉字太少。1995 年的汉字扩展规范 GBK 1.0 收录了 21 886 个符号,它分为汉字区和图形符号区。汉字区包括 21 003 个字符。2000 年的 GB 18030 是取代 GBK 1.0 的正式国家标准。该标准收录了 27 484 个汉字,同时还收录了藏文、蒙文、维吾尔文等主要的少数民族文字。

Unicode 也是一种字符编码方法,它是由国际组织设计、可以容纳全世界所有语言文字的编码方案。Unicode 的全称是"Universal Multiple-Octet Coded Character Set",简称为 UCS。UCS 可以看做是"Unicode Character Set"的缩写。Unicode 是两字节的全编码,无论是汉字还是 ASCII 字符都使用两字节表示。早期的 VB 采用单字节处理机制,也称为 ANSI 处理方式;到了 VB 4.0 以后,VB 开始使用 Unicode 方式来处理字符,即每个汉字、每个英文字符都看做一个字符,但所占用的空间都是两个字节。

由于 ANSI 方式与 Unicode 方式两者对汉字的处理方式是不同的,这样在处理汉字时会带来混乱。例如,字符串"VB 程序设计"在 ANSI 方式中的长度是 10,而在 Unicode 方式中却是 6。为了解决这个问题,VB 从 5.0 以后新增加了一个 StrConv()转换函数,可以将字符串在 ANSI 与 Unicode 两种方式之间转换。另外,新版的 VB 还增加了一套专门用于处理字符存放字节的函数,即在原有的字符串处理函数后加上一个"B"。

例如:

```
str1 = "VB 程序设计"
Print Len(str1)                    '返回 6
Print LenB(str1)                   '返回 12
```

Print 是一个输出命令,用于在指定对象上输出其后的表达式的值,该命令常用于测试某个表达式或函数的值。在立即窗口中可以直接使用该命令,也可以用"?"来代替 Print 命令。

2. 字符串函数

VB所提供的字符串处理函数非常丰富,使用十分灵活方便,这也是VB的特点之一。VB中的字符串函数大多以可以省略的类型说明符“$”结尾,表示函数的返回值为字符串类型。本节在介绍时将全部省略其类型符“$”。

(1) 删除空格函数

VB提供了3种不同的删除空格函数:LTrim()、RTrim()和Trim(),其参数均为字符串表达式,功能分别为将指定字符串的左边、右边和左右的空格删除。

(2) 截取子串函数

截取子串是指在一个指定的字符串表达式中取出一个指定的子串,可以从字符串的左边、右边和中间来取子串,因此有如下3种对应的取子串函数。

- Left(C,n):返回字符串C左边n个字符的子串。
- Right(C,n):返回字符串C右边n个字符的子串。
- Mid(C,p,n):返回字符串C中从位置p开始的n个字符。

例如:

```
? Left("Visual",3)                    '返回“Vis”
? Right("Visual",3)                   '返回“ual”
strV = "Visual Basic"
? Left(strV,1) + Mid(strV,8,1)        '返回“VB”
```

(3) 字符串长度测试函数

前面已经介绍,在Unicode方式下,测试一个字符串的长度有两个函数:Len()与LenB(),分别用于测试字符串中的字符个数和字符串在内存中存放的字节数。如果没有使用StrConv()函数进行转换,LenB()函数的返回值正好是Len()函数返回值的两倍。

例如:

```
? Len("China! 中国")                  '返回8
? LenB("China! 中国")                 '返回16
```

(4)字符串匹配函数

字符串匹配函数用来在指定字符串中查找某个字符串,如在姓名文本框的输入中查找第一个字是否为“马”。InStr函数就可以实现这样的功能,格式为:

```
InStr([p1,]C1,C2)
```

其功能是:在字符串C1中从p1位置开始查找字符串C2,如找到则返回C2中第一个字符在C1中的位置。如省略p1则从C1的第一个字符开始查找。

```
? InStr("China! 中国","中国")         '返回7
? InStr("China! 中国","美国")         '返回0
```

其他的字符串处理函数都比较简单,使用时可参见表3-3中的说明或查阅联机手册。

3.3.3 Shell函数

在VB程序中,有时需要调用外部程序或命令,例如在程序中调用Windows的“计算器”程序。Shell()函数可以满足这样的要求,其格式为:

```
Shell(pathname[,windowstyle])
```

其中，pathname 为需要执行的程序名，可以带路径；windowstyle 表示程序运行窗口的风格，其取值如表 3-4 所示。

表 3-4 windowstyle 取值

VB 常量	值	含 义
vbHide	0	窗口被隐藏，且焦点会移到隐式窗口
VbNormalFocus	1	窗口具有焦点，且会还原到它原来的大小和位置
VbMinimizedFocus	2	窗口会以一个具有焦点的图标来显示
VbMaximizedFocus	3	窗口是一个具有焦点的最大化窗口
VbNormalNoFocus	4	窗口会被还原到最近使用的大小和位置，而当前活动的窗口仍然保持活动
VbMinimizedNoFocus	6	窗口会以一个图标来显示，而当前活动的窗口仍然保持活动

该函数具有返回值：如果调用成功，则返回调用程序的 ID 号；如不成功则返回 0。

3.4 输入与输出

3.4.1 数据的输入

VB 提供了两个函数用来与用户交互，其中之一就是用于数据输入的函数 InputBox()。

InputBox 函数用来接收键盘的输入，例如以下的语句将显示如图 3-3 所示的输入框：

```
strName = InputBox("请输入要查找的姓名:","输入")
```

其典型使用格式为：

```
InputBox(prompt[, title] [, default])
```

prompt 是输入框中的提示文字，如“请输入要查找的姓名”；title 为输入框的标题，如“输入”；default 为显示在输入文本框中的默认内容，省略则显示空串。

从图 3-3 可以看出，该函数的作用是打开一个对话框，等待用户输入信息，信息输入完成后单击“确定”按钮或按 Enter 键，函数将返回文本框中输入的字符型值并赋给指定的变量。如在图 3-3 中输入“张三”后按 Enter 键，则 strName 变量中的内容即为“张三”。

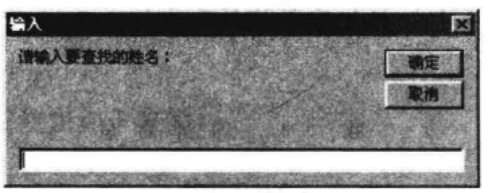


图 3-3 信息输入框

在使用该函数时，除了第一项提示外都是可以省略的，但要注意如果省略一部分参数，被省略的参数必须用占位符——逗号跳过。

该函数在使用时要注意以下几点：

① 提示文字的最大长度为 1 024 个字符，在对话框中显示这些提示信息时会自动换行。如果想按自己的要求换行，必须在提示信息中插入回车换行来实现。例如，将提示信息写成：“请输入要查找的姓名" + Chr(13) + Chr(10) + "单击确定按钮或按 Enter 键结束”。执行后将在对话框中显示两行文字，其中的 Chr(13) + Chr(10)也可以用 VB 的常量“vbCrLf”来代替。

② 在默认情况下，该函数的返回值为字符串型，如果用户没有输入而直接按 Enter 键，则会

返回空字符串。

③ 如果需要接收一个数值(如年龄、成绩等),则通常要将接收返回值的变量声明为一个数值类型的变量。如果事先没有声明则该变量仍然是一个字符型,因此在使用该变量前一般要进行转换以防止类型不匹配的错误。另外,如果已将该变量声明为数值型变量,为防止出现运行错误,一般要设置 default 参数以保证用户不输入也有一个默认的数值返回到变量中。

④ 该函数在运行时还有一个"取消"按钮,用于将输入的值作废,相当于按 Esc 键,此时将返回一个空串。

⑤ 每次调用该函数只能输入一个值,如果需要连续输入多个值,必须将该函数与循环语句联合使用,每调用一次将输入的数据保存下来后(通常是存放到一个数组中)再次调用。

3.4.2　数据的输出

在 VB 中数据的输出有多种方法,最常用的有 3 种:使用 MsgBox()函数或命令、使用 Print 方法和利用窗体设计输出界面。本节仅讨论前两种方法,第三种方法将在以后的章节中陆续介绍。

1. MsgBox()函数或命令

在 Windows 及其应用软件中,消息框的使用非常频繁,像警告信息、提示信息、确认信息等均通过消息框显示。VB 也提供了消息框功能,如图 3-4 所示的询问消息框就是用 MsgBox 函数实现的。

MsgBox()函数的基本格式如下:

```
MsgBox(prompt[, buttons] [, title])
```

其中,prompt 指消息框中的提示信息,如图 3-4 中"继续吗?";buttons 定义消息框中的按钮类型、数目及图标样式;title 即消息框的标题,如图 3-4 中的"询问"。

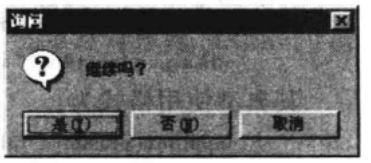


图 3-4　询问消息框

应用 MsgBox()函数时最主要的是确定第二项参数,该参数共有 4 组,即为 4 项参数之和:"按钮数目值" + "图标类型值" + "默认按钮值" + "模式值",有两种取法,一种是直接用 4 个取值相加,另一种是用 4 个内部常量相加。按钮参数值的取法如表 3-5 所示。

表 3-5　buttons 设置值及意义

参　数	内部常量	取　值	描　述
按钮数目	VbOKOnly	0	只显示确定按钮(默认)
	VbOKCancel	1	显示确定与取消按钮
	VbAbortRetryIgnore	2	显示终止、重试与忽略按钮
	VbYesNoCancel	3	显示是、否与取消按钮
	VbYesNo	4	显示是与否按钮
	VbRetryCancel	5	显示重试与取消按钮
图标类型	VbCritical	16	显示关键信息图标(红色 STOP 标志)
	VbQuestion	32	显示询问信息图标(?)
	VbExclamation	48	显示警告信息图标(!)
	VbInformation	64	显示普通信息图标(i)

续表

参　数	内部常量	取　值	描　述
缺省按钮	VbDefaultButton1	0	第一个按钮为默认按钮
	VbDefaultButton2	256	第二个按钮为默认按钮
	VbDefaultButton3	512	第三个按钮为默认按钮
模式	VbApplicationModel	0	应用模式(默认)
	VbSystemModel	4 096	系统模式

图 3-4 的消息框可用下列命令之一实现：

```
intRet = MsgBox("继续吗?",VbYesNoCancel + VbQuestion,"询问")　'用内部常量
intRet = MsgBox("继续吗?",3 + 32,"询问")　'用取值
```

如果 prompt 的内容较多(最长亦为 1 024 个字符)需要换行时，不能直接按 Enter 键，必须使用 Chr(13) + Chr(10)来实现。

命令执行后，函数有一个返回值送到变量 intRet 中，返回值取决于用户响应了哪一个按钮，各个按钮对应的返回值如表 3-6 所示。程序可以根据用户响应的按钮所返回的值决定程序的流程。

表 3-6　MsgBox()函数返回值

响应按钮名	内部常量	返　回　值
确定	VbOk	1
取消	VbCancel	2
终止	VbAbort	3
重试	VbRetry	4
忽略	VbIgnore	5
是	VbYes	6
否	VbNo	7

MsgBox 也可以以命令的形式来使用，其语句格式可以使用如下格式：

```
MsgBox prompt[, buttons] [, title]
```

这种方式只有在不需要返回用户所做的选择的情况下方可使用，所以一般都使用其函数形式，以捕捉用户的选择并对程序的流程做出控制。

无论是以函数的方式还是以命令的方式使用 MsgBox，该消息框显示后有一个共同的特点，就是在出现消息框后用户必须做出相应的选择，否则不能进行其他的操作。在 VB 中，这样的窗口称为“模态窗口”，反之称为“非模态窗口”。

2. Print 方法

为了将文本或运行结果输出到屏幕或打印机上，可以使用 Print 方法。该方法的使用格式为：

```
[object].Print [{Spc(n) | Tab(n)} expression][,|;]
```

其中,object为输出的目标,可以是立即窗口、指定窗体、图片框或打印机。如果是立即窗口,用"Debug",如果是窗体或图片框,则是窗体或图片框的名称。如该项省略,则将结果输出到当前窗体。expression为输出的字符表达式或数值表达式列表,各表达式之间的分隔用";"或","。当采用","分隔时,前后的输出项分别输出到两个标准分区,每个分区的长度为14个字符宽;如采用";"分隔时,前后两个输出项将紧凑输出。

默认情况下,每执行一个Print方法都要自动换行。为了能在同一行上输出下一个Print方法的内容,可以在末尾加上","或";",其含义同上。

Spc(n)表示输出空格,n为空格数。Tab(n)要求将结果输出到指定的列上,n为列号。

例如:

```
Debug. Print "欢迎来到奇妙的 VB 世界!";"您好!"          '在立即窗口输出
Form1. Print "2 +3 = ";2 +3                              '在指定窗体上以紧凑格式输出
Print                                                    '输出一个空行
Print Tab(12),"Welcome to VB!"                           '在当前窗体的第 12 列输出
```

3.5 程序流程控制

利用VB的集成开发环境开发应用程序时,主要完成两方面的工作:一方面是进行用户界面的设计,另一方面更重要的是根据不同的事件来编制不同的程序代码以完成预定的功能。VB的程序代码与一般的程序设计语言一样,也具有3种基本结构:顺序结构、判定结构和循环结构。顺序结构是指程序按代码的书写顺序依次执行,下面着重介绍判定结构和循环结构。

3.5.1 判定结构

1. 条件结构

条件结构语句分单行结构与块结构两种形式。

(1) 单行结构

```
If condition Then statement[ Else elsestatement]
```

(2) 块结构

```
If condition Then
    [statements]
[ElseIf condition - n Then
    [elseifstatements] ...
[Else
    [elsestatements]]
End If
```

单行结构适用于简单的测试,块结构不仅具有更强的结构性与适应性,而且也易于维护与调试。单行结构与块结构的区别有两点:

① 块结构的Then后面没有语句,而单行结构的肯定有语句。

② 单行结构没有End If结束语句,而块结构必须有该结束语句。

两种结构的流程如图3-5和图3-6所示。

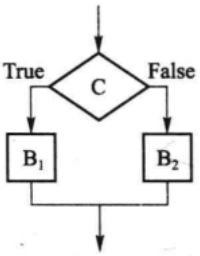


图 3-5 单行结构流程

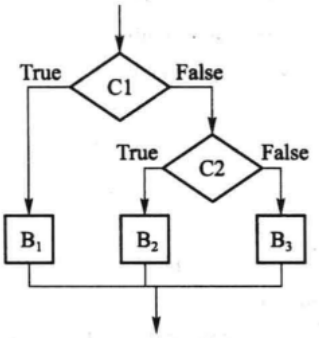


图 3-6 块结构流程(三分支)

在单行结构中,condition 是条件表达式,如果 condition 为 True,则执行 Then 后的语句,否则执行 Else 后的语句。如果省略 Else 语句,则直接执行下一条语句。

例如,将 x, y 两数中的较大值送到 z 中的单行判定结构可写成:

```
if x < y then z = y else z = x
```

块结构可以实现多重判定,即有多个条件。从第一个条件开始逐个测试,条件为 False 再测试下一个条件。当找到一个条件为 True 时,即执行相应的语句块,然后执行 End If 后面的语句。如果所有条件都不是 True,则执行 Else 后面的语句。

【例 3-3】 根据学生的成绩显示不同的评语。

程序代码如下:

```
sngCj = InputBox("请输入学生的成绩:", "输入成绩")
If sngCj >= 85 Then
    MsgBox ("成绩优秀")
ElseIf sngCj >= 60 And sngCj < 85 Then
    MsgBox ("成绩合格")
Else
    MsgBox ("成绩较差")
End If
```

在 Vbxx 工程中新建一个名为 frmExam_3_3 的窗体,在窗体的 Click 事件中输入上述代码。

2. Iif()函数

VB 还提供了与单行结构的判定语句相对应的 IIf 函数,是单行判定结构的简写形式,其格式如下:

```
IIf(condition, truepart, falsepart)
```

其中,condition 为条件。如果条件为真(True),则返回 truepart 表达式的值;反之返回 falsepart 表达式的值。利用 IIF()函数可使程序大为简化,如上面的将 x, y 中的较大值送到 z 中用 IIF()函数则会更为简捷:

```
z = IIf(x < y, y, x)
```

3. 选择结构

有时需要判定的条件较多,如果再用以上的条件结构,则代码显得十分繁琐且容易出错。为此,VB 提供了专门用于多重判定的选择结构(也称为情况语句)。

选择结构的语法是:

```
Select Case testexpression
    [Case expressionlist1
      [statementblock - 1]]
    [Case expressionlist2
      [statementblock - 2]]
          ...
    [Case Else
      [statementblock - n]]
End Select
```

其中,testexpression 为测试表达式,每个 Case 后的 expressionlist1 为表达式列表。VB 在执行时首先计算测试表达式 testexpression 的值,然后将该值与每个 Case 后的值进行比较,若值相同就执行该 Case 后的语句块。如果有多个 Case 值与之相匹配,只执行第一个;如果没有一个匹配,则执行 Case Else 后的语句块。

【例 3-4】 例 3-3 用选择结构后即变成以下代码,其执行效果完全相同。

```
sngCj = InputBox("请输入学生的成绩:","输入成绩")
Select Case sngCj
    Case 85 To 100
        MsgBox ("成绩优秀")
    Case 60 To 84
        MsgBox ("成绩合格")
    Case Else
        MsgBox ("成绩较差")
End Select
```

在 Vbxx 工程中新建一个名为 frmExam_3_4 的窗体,在窗体的 Click 事件中输入上述代码。

使用选择结构语句的关键是选择合适的测试表达式,并确定合适的测试值。测试值的表示方法可以有以下几种:

- 具体取值,可以是一个或多个,如 100, 85, 1。
- 连续的数据范围,如上例 85 To 100、60 To 84 等。
- 利用 Is 关键字来判定,如对于成绩而言不可能大于 100,因此 85 To 100 就可以写成 Is > =85。
- 以上几种表示方法的组合,如 100, 60 To 84, Is > =85。

3.5.2 循环结构

循环结构也是程序的一种基本结构,VB 提供了 3 种不同风格的循环结构,即 Do 循环、For 循环和 For Each 循环(For Each 循环详见第 9 章)。

1. Do 循环

Do 循环主要用于循环次数预先不能确定的循环结构。它有两种语法形式:当型与直到型。

(1) 当型循环

```
Do [{While | Until} condition]
```

```
        [statements1]
        [Exit Do]
        [statements2]
    Loop
```

(2) 直到型循环

```
    Do
        [statements1]
        [Exit Do]
        [statements2]
    Loop [{While | Until} condition]
```

当型循环是先判断,后执行,循环体中的语句可能一次也不执行。而直到型循环是先执行后判断,循环体中的语句至少执行一次。两种语法(While 关键字)的流程图如图 3-7 和图 3-8 所示。

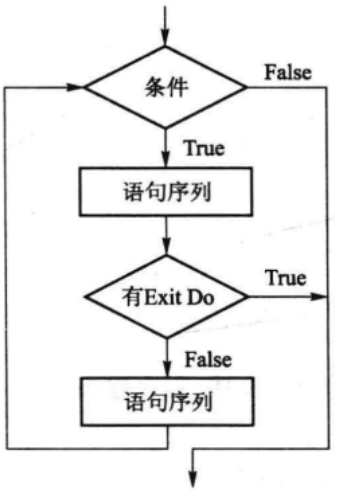


图 3-7 当型循环

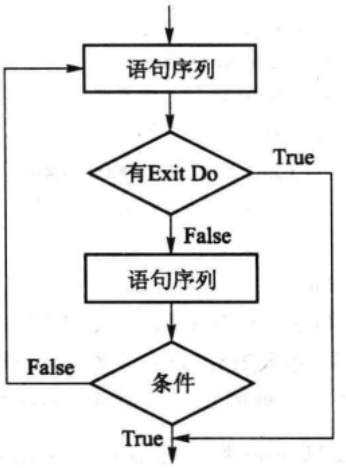


图 3-8 直到型循环

两种循环均是通过关键字 While 和 Until 来控制循环执行的条件的,关键字 While 表示条件为真时循环执行,而 Until 则正好相反。

Exit Do 语句执行时将强制退出循环,这一句是可选的,它一般与判定结构的语句联合使用,以免造成死循环。

【例 3-5】 在例 3-4 中,只能判断一次成绩,现在利用循环将其改成反复输入,直到按 Enter 键结束。程序代码如下:

```
Do While True
  sngcj = InputBox("请输入学生的成绩:", "输入成绩")
  If sngcj = "" Then
      Exit Do
  Else
      Select Case sngcj
          Case Is >= 85
              MsgBox ("成绩优秀")
```

```
            Case Is >= 60
                MsgBox ("成绩合格")
            Case Else
                MsgBox ("成绩较差")
        End Select
    End If
Loop
```

在Vbxx工程中新建一个名为frmExam_3_5的窗体,在窗体的Click事件中输入上述代码。

思考: 该程序采用了Do While...Loop的形式,能否改成其他两种形式?

【例3-6】 假设目前世界人口为60亿,如果每年以1.4%的速度增长,问多少年后世界人口达到70亿。程序代码如下:

```
Dim p As Double
Dim r As Single
Dim n As Integer
p = 6000000000#
r = 0.014
n = 0
Do Until p >= 7000000000#
    p = p * (1 + r)
    n = n + 1
Loop
Print n; "年后"; "世界人口达"; p
```

将代码输入frmExam_3_6的窗体的Click事件过程中,单击窗体,程序输出:

```
12 年后世界人口达 7089354809.76375
```

思考: 该程序如果采用Do...Loop Until的形式,程序该如何修改?

VB还提供了比该循环稍简单一点的While循环,其语法格式如下:

```
While condition
    [statements]
Wend
```

从语法结构上看,这种循环没有Exit Do关键字,不能强行退出循环,因而其控制功能比Do循环要稍差一些,但使用却更简单了。

2. For循环

Do循环一般用于循环次数未知的情况。当对于循环次数已知时,最好采用For循环。与Do循环不同,For循环含有一个循环计数变量,每执行一次循环,该变量就会增加或减少指定的值,直到循环结束。其语法格式如下:

```
For counter = start To end [Step stepl]
    [statements]
    [Exit For]
    [statements]
Next [counter]
```

该语句执行时,先将 counter 设为 start(初值)。测试 counter 的值是否超过 end(如 stepl 为正数是大于,否则是小于),若已经超过,则循环结束,否则执行循环体中的语句。stepl 是每次循环时 counter 变化的数值,它可正可负,默认值为 1。本次循环结束后,counter 将加上步长 stepl 的值返回到循环开始,再进行测试,直到 counter 的值超过终值,循环结束。同样可以在循环体内加上判定结构与 Exit For 结合从而提前退出循环。

不要在循环体内修改循环计数变量的值,否则会造成循环次数的不准确,而且程序调试也非常困难。

初值、终值与步长均为数值表达式,但其值不一定是整数,也可以是实数,VB 会自动取整。

【例 3-7】 统计一个成绩数组中的平均分数。

分析:首先要定义一个数组,数组的大小可以在运行时确定,即为动态数组。然后通过一个循环累加每个学生的成绩,最后即可计算出平均成绩。

```
Dim i, j, intRs As Integer
Dim strMsg As String
Dim sngCj( ), sngAver As Single                    '定义动态数组
intRs = InputBox("请输入学生数目:", "学生数")      '输入学生的数目以决定数组的大小
ReDim sngCj(intRs + 1)                             '重新定义数组
sngAver = 0
For i = 1 To intRs
    strMsg = "请输入第" & Str(i) & "个学生的成绩:"
    sngCj(i) = InputBox(j, "成绩")                 '利用字符变量动态改变提示的内容
    sngAver = sngAver + sngCj(i)
Next
sngAver = sngAver / intRs
Print "该门课程的平均成绩为:", sngAver
```

在 Vbxx 工程中新建一个名为 frmExam_3_7 的窗体,在窗体的 Click 事件中输入上述代码。

3. 循环的嵌套

无论是 Do 循环还是 For 循环,在其循环体内都可以嵌套另一层循环,这种大循环套着小循环的情况称为循环的嵌套。在编写循环嵌套的程序时,最重要的是小循环一定要完整地包含在大循环中,内外循环绝对不能交叉。

【例 3-8】 数组的排序。

数据排序的算法问题在任何程序设计语言中都作为一种典型的算法来介绍,因此本书不再赘述。程序代码如下:

```
Dim i, j, intMin As Integer
Dim strMsg As String
Dim sngSq(5), sngT As Single
'输入原始数据
For i = 1 To 5
    strMsg = "请输入第" & Str(i) & "个数:"
```

```
        sngSq(i) = Val(InputBox(j, "数据"))          '将输入的数据转换为数值型
    Next
    '排序
    For i = 1 To 4
        intMin = i                                   '对第 i 遍排序,初始假定第 i 个元素最小
        For j = i + 1 To 5                           '找出最小的元素序号
            If sngSq(j) < sngSq(intMin) Then intMin = j
        Next j
        '将找出的最小元素与第 i 个元素交换
        sngT = sngSq(i)
        sngSq(i) = sngSq(intMin)
        sngSq(intMin) = sngT
    Next i
    '打印输出
    For i = 1 To 5
        Print sngSq(i);                              '不换行输出
    Next i
```

在 Vbxx 工程中新建一个名为 frmExam_3_8 的窗体,在窗体的 Click 事件中输入上述代码。

【例 3-9】 输出指定两个整数之间的所有素数。所谓素数是指除了 1 与其自身外不能被任何数整除的数。程序代码如下:

```
Private Sub Form_Click()
m = InputBox("请输入一个整数")
n = InputBox("请输入另一个整数")
If n < m Then
    t = n
    n = m
    m = t
End If
For i = m To n
    k = Int(Sqr(i))
    j = 2
    swit = 0
    While j <= k And swit = 0
        If i Mod j = 0 Then
            swit = 1
        Else
            j = j + 1
        End If
    Wend
    If swit = 0 Then
        d = d + 1
        If d Mod 5 = 0 Then
```

```
                Print i; "   ";
                Print
            Else
                Print i; "   ";
            End If
        End If
    Next i
    End Sub
```

将程序保存在 frmExam_3_9 的窗体中,在输入 100 与 300 后,运行的结果如图 3-9 所示。

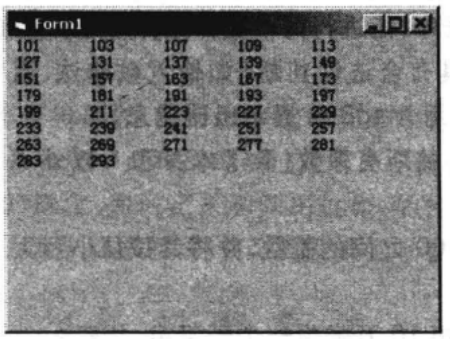


图 3-9 输出 100~300 之间的素数

习 题 3

1. VB 提供了哪几种标准数据类型? 在声明类型时分别使用什么关键字? 其类型符又是什么?

2. 在 VB 中有哪些模块? 几种模块有哪些区别? 如何使用?

3. 在代码窗口中编程时要注意哪些问题?

4. 变量在命名时应遵守哪些规则?

5. 如何定义并使用数组?

6. VB 提供了哪几种人机交互函数? 如何使用? MsgBox 过程与 MsgBox()函数的区别是什么?

7. 控制程序结构的语句有哪些? 两种循环结构如何使用?

8. 编程中可能出现的错误有哪些? 如何调试程序中出现的错误?

9. 下列符号名中,哪些是合法的变量名?

 bolHf A5h intNl 5. num 123abc X_YZ X-Y a$b

10. 下列数据中哪些是合法的常量?

 100 &O100 &H100 'ABC' "ABC" ABC$ #02/25/1999# "02/25/1999"

 100! 1E+2 #January 1,1999# Yes No true False

11. 写出下列对应的 VB 表达式。

(1) $\dfrac{a}{2+\dfrac{x}{y}}$　　(2) $\dfrac{2y}{(ax+by)(ax-by)}$

(3) $\sqrt{\ln\dfrac{x}{y}}$　　(4) $X+Y$小于10,且$X-Y$要大于0。

(5) A、B之一为0,但不能同时为0。

(6) 性别不是女且出生日期在1966年1月1日前或性别不是男且职称不是高工。

12. 利用InputBox()函数输入三角形三边的长度a、b、c,计算三角形的面积并显示在窗体上。

13. 编制一个模拟计算程序,能进行两个操作数算术运算,要求窗体如图3-10所示。同时能对输入的操作数及运算符具有合法性判断(如输入数非法、运算符非法、除数为0等),并利用MsgBox显示出错信息。

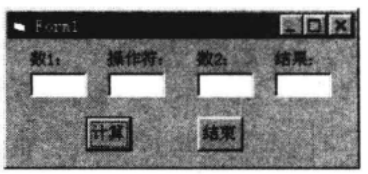


图3-10　模拟计算程序窗体

14. 求一个指定整数以内的所有素数(除了本身和1以外不能被任何数整除的数)。

15. 随机产生10个0~100之间的整数,并将其按从小到大的顺序显示在窗体上。

16. 模拟摇奖程序:规定每10万张为一个开奖组,共有4个开奖组(组号为A~D),奖券号码为组号加一个5位数码,如A40235表示A组中的号码为40235的奖券。每组的中奖数目由键盘输入,用程序来随机产生相应的中奖号并打印在窗体上。

17. 设计一个口令加密程序,口令由字母构成,将输入的每个字母循环左移两个字母(区分大小写),如下表所示。

明文	A	B	C	…	X	Y	Z
密码	Y	Z	A	…	V	W	X

程序运行后的界面如图3-11所示。

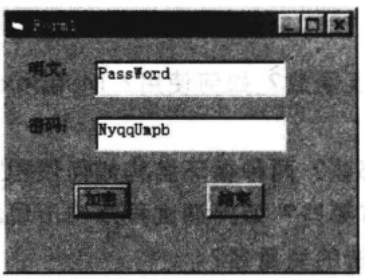


图3-11　题17程序运行窗体

第4章　VB中的过程

学习目标

掌握VB中过程的编制方法，熟悉事件过程、窗体过程、通用过程以及函数过程的应用，加深对VB事件驱动机制的理解。对变量的作用范围及过程调用中的参数传递有较深入的理解，了解VB中的面向对象程序设计的基本原理与方法。

过程是VB应用程序的基本逻辑部件，任何程序代码都必须包含在某一个过程之中。通过前面几章的学习，读者对VB应用程序中过程的编制已经有所了解（如编制一个命令按钮的单击事件过程代码），但对诸如过程的概念、如何定义和调用过程、如何将参数传递到过程等概念还缺乏系统的认识。本章正是为了解决这些问题，使读者能真正掌握VB中的过程。

4.1　过程概述

所谓过程是指能完成某种特定功能，且能被反复调用的一组程序代码。VB中的过程可以分为两大类，一类是VB本身所提供的大量内部函数过程和事件过程；另一类是用户根据需要自己定义的、可供事件过程反复调用的自定义过程。事件过程构成了应用程序的主体，而自定义过程则是为事件过程服务，供事件过程调用的。

在VB中自定义过程又可以分以下4种。

① Sub过程：以Sub为关键字的过程，也称为子过程。实际上，通常的事件过程也是Sub过程。

② Function过程：以Function为关键字的过程又称为函数。

③ Property过程：以Property为关键字的过程称为属性过程，主要用于用户自定义控件的属性。

④ Event过程：以Event为关键字的过程称为事件过程，主要用于用户自定义事件。

对于VB过程有以下几点说明：

① VB中的任何过程均必须包含在某一个模块之中，要么是窗体模块，要么是标准模块。

② Sub过程在调用后不会返回值，而Function过程会返回一个值。

③ 用户自定义过程根据其作用范围的不同可分为公有过程和私有过程两种。

④ 事件过程的作用范围一般局限于该模块范围之内。

本书限于篇幅，只讨论前两种过程。

4.2　过程定义与调用

4.2.1　过程的定义与建立

1. Sub过程

Sub过程分为事件过程与自定义过程两大类，两类过程的定义方式基本相同。定义的语法如下：

```
[Private | Public  Sub name [(arglist)]
        [statements]
[Exit Sub]
        [statements]
End Sub
```

Private | Public 表示过程的作用范围,Private 表示所定义的过程是模块级过程(私有),只能被本窗体模块内的过程所调用,对事件过程来说,只能用 Private。Public 表示所定义的过程是全局的或公用的,它可以被应用程序内任一过程所调用。

name [(arglist)]为过程名及参数列表。通用过程名的命名规则与变量命名相同,而对于事件过程来说,由于所属的对象名及事件均已经确定,因此不需再命名,系统会自动命名。例如:窗体的 Load 事件过程的过程名是 Form_Load(),一个名为 Command1 的命令按钮的 Click 事件的过程名为"Command1_Click()"。

arglist 为参数列表,表示过程可以接受的参数,这种参数称为形式参数(简称形参)。形参可以是变量名或数组名,如有多个参数,要用逗号隔开。当然也可以不带参数,但括号不能省略。形式参数的格式是:[ByVal | ByRef] varname[()] [As type]。

[ByVal | ByRef]表示参数的传递方式:ByVal 为按值传递,ByRef 为按地址传递,默认为 ByRef。varname[()]为变量名或数组名,如有括号则表示该变量为数组,否则为变量。[As type]表示形参的数据类型,如省略则为 Variant 型。



事件过程中形式参数的个数、位置及类型等完全是由具体事件本身决定的,用户不可随意增减。

Exit Sub 表示立即从一个 Sub 过程中退出,程序继续执行调用该过程语句的下一条语句,显然事件过程中一般不能出现该语句。

事件过程的建立方法与通用过程的建立方法不太一样。

(1) 建立事件过程的具体步骤

① 打开要编写事件过程的窗体。

② 双击需编写事件过程的对象(窗体或控件),即打开相应的代码编辑器窗口,同时在窗口中将显示默认的事件过程模板。对窗体而言,其默认事件过程为 Form_Load(),而对于按钮默认事件过程为 Command1_Click()。



在选定对象上右击选择"查看代码"或选择[视图]→[代码窗口]菜单命令均可进入代码编辑器窗口。

③ 在"过程"列表框中选择事件过程,在代码窗口中也会显示选定的事件过程模板。如要编写窗体的 Activate 事件过程,屏幕显示如图 4-1 所示。

④ 在 Private Sub Form_Activate()与 End Sub 之间输入代码。

⑤ 输入完成后运行调试,如无错误即可保存。

(2) 自定义过程的建立

自定义过程既可以放在窗体模块中,也可以放在标准模块中,但都必须放在"通用"段中。自定义过程的建立方法有两种。第一种方法为交互式,以在窗体中添加一个名称为 MySub 的自

定义过程为例,采用交互式的步骤如下:

① 打开需定义过程所在的代码编辑器窗口。

② 选择〖工具〗→〖添加过程〗菜单命令,出现如图 4-2 所示的“添加过程”对话框。

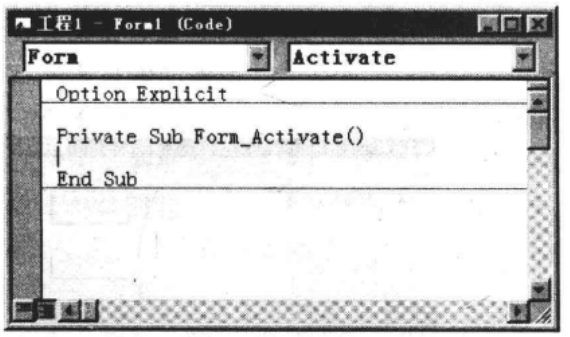


图 4-1 事件过程的创建

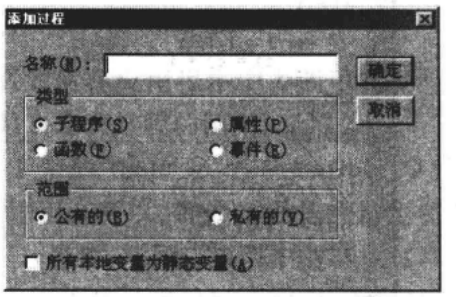


图 4-2 “添加过程”对话框

③ 在名称框中输入“MySub”,选定过程类型为“子程序”,范围为“公有的”,单击“确定”按钮。

④ 代码编辑器窗口中同样会出现相应的 MySub 过程模板,在模板中即可输入相应的代码。

过程的这种建立方法没有考虑形参,形参还必须手工输入。

建立过程的第二种方法是直接输入法。先在代码编辑器窗口的“对象”列表框中选择“通用”,再在代码区的空白处直接输入过程定义的第一句,按 Enter 键后也会出现相应的过程模板。

【例 4-1】 编写一个求 n! 的过程。

新建一个窗体,其上放置如表 4-1 所示的对象并设置相关的属性。

表 4-1 属 性 设 置

类 型	属 性 名 称	属 性 值
标签	Caption Name	请输入数据 N: lblsj
标签	Caption Name	N 的阶乘为: lblcj
文本框	Name Text	TXTSJ 0
文本框	Name Text	TXTJCJ 1
命令按钮	Caption Name Default	求阶乘 cmdjc True
命令按钮	Caption Name	关闭 cmdclose

以上窗体及相关的控件设计完成后,打开窗体的代码窗口,输入如图 4-3 所示的代码。运行该程序,其结果如图 4-4 所示。



图4-3 求阶乘的过程代码

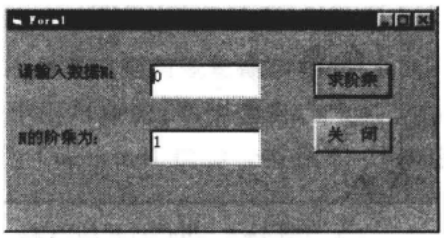


图4-4 运行界面

运行无误后,将窗体以 frmExam_4_1 为名保存在 Vbxx 工程中。

2. Function 过程

上一章已经学习了 VB 所提供的可直接在程序中使用的诸多内部函数,除了这些内部函数以外,VB 还允许用户像定义过程那样自己定义函数,这就是 Function 过程。事实上在 VB 中,Sub 过程与 Function 过程并无太大的差别,这从两者的名称也可以看出。

定义 Function 过程的主要语法如下:

```
[Public | Private] Function name [(arglist)] [As type]
    [statements]
    [name = expression]
    [Exit function]
    [Statements]
    [name = expression]
End Function
```

从语法格式上看,Function 过程的定义与 Sub 过程的定义大体相似,不同点有两个:

① 既然是函数当然要有返回值,返回值的类型用 As type 来指明,如取默认值,则函数的类型为 Variant 类型。

② 函数的返回值通过 name = expression 格式来赋值。如取默认值,该过程会返回对应类型的默认值。如数值函数则返回值 0,字符串函数返回空字符串等。

函数的建立方法与通用过程的建立方法完全相同。

思考: 例 4-1 中的求阶乘运算如用函数过程来实现,应如何编写程序代码?

4.2.2 过程的调用

1. 实参与形参

形参是指出现在过程定义的名称后的括号中的变量名或数组名,在过程未被调用之前,系统并没有为其分配实际的内存,仅仅是形式上的名字。

实参是在调用过程时,传递给相应过程的实际数据,它们可以是变量名、数组名、常数或表达式等,相应的过程在对这些数据进行处理后,将处理的结果再通过适当的形式返回给调用程序。

实参与形参在名字上并不要求相同,它们是按位置来匹配的。因此要求实参的个数、类型和位置等与形参必须一致。

2. Sub 过程的调用

在例 4-1 中,定义并建立了一个求阶乘的过程,那么如何在应用程序中去调用它呢? Sub 过程的调用分两种情况:一种是事件过程的调用,另一种是自定义过程的调用。

对于事件过程显然必须是由事件来驱动的,如例 4-1 中的"求阶乘"按钮的 Click 事件过程,只有在用户单击了该按钮后才能调用 Cmdjc_Click()过程。所以,事件过程是由发生在 VB 应用程序中的事件来自动调用的,这种调用也叫隐式调用。当然,对同一模块中的其他过程,也可以显式地调用事件过程。而自定义过程必须在事件过程或其他过程中显式调用,否则过程的代码永远也得不到执行。自定义过程的调用是通过 Call 语句来实现的。

利用 Call 语句调用自定义过程的格式如下:

```
[Call] name [(argumentlist)]
```

其中,Call 为命令关键字,可以省略。如该关键字不省,则其后的实参 argumentlist 必须加上括号;如该关键字省略,实参 argumentlist 的括号也可以省略。name 为调用的过程名称,argumentlist 为实参列表。如果所建立的过程是有参过程,实参的个数、类型与顺序应与被调用过程的形式参数相匹配,各参数之间同样用逗号分开。如果是一个无参过程,则调用时括号也可省略。如果形式参数是一个数组,则调用时只要给出数组名与空的括号即可。

对例 4-1 所定义的过程可以采用两种方法来调用:

```
sub_jc intA, lngB
Call sub_jc (intA, lngB)
```

Call 语句不仅可以调用 Sub 过程,也可以调用 Function 过程及动态链接库中的过程。

3. Function 过程的调用

由于 Function 过程实际上就是一个用户自定义函数,因此使用 Function 过程的方法与使用 VB 内部函数的方法是一样的,即将 Function 过程名及相应的实参放在表达式中。但调用 Function 过程时即使是无参过程,其括号也不能省略。

另外,也可以像调用 Sub 过程那样调用 Function 过程,但这种调用将得不到函数的返回值。

例如:例 4-1 中的求阶乘的过程用函数过程来编写,代码如下:

```
Public Function fun_Jc(ByVal x As Integer)
    Dim i As Integer
    Dim cj As Long
    cj = 1
    If x > 0 Then
      For i = 1 To x
        cj = cj * i
      Next i
    End If
```

```
    fun_jc = cj
End Function
```

则调用该函数过程用下列语句之一均可以实现：

```
txtJcj. Text = fun_Jc(txtSj. Text)
Call fun_Jc (txtSj. Text)
fun_Jc txtSj. Text
```

后两个语句在调用时，VB将放弃返回值。由上面的分析可以看出，对于同一个问题，有时可以用Sub过程来解决，也可以用Function过程来解决。

4.3 过程调用中的参数传递

无论是Sub过程还是Function过程都可以带有形式参数，在调用这些过程时首先要“形、实结合”，以实现调用程序与被调用过程之间的数据传递。这种数据传递必须解决两个问题：一是如何将待处理的数据传送给相应的过程，二是如何将过程处理后的结果带回到调用程序。为此，VB提供了两种参数传递方法：按值传送与按地址传送。

4.3.1 按地址传送

按地址传送通过在过程定义时在形参前加上ByRef来标识，这也是一种默认的传送方式。按地址传送的形参与实参这样结合：调用过程时，系统将实参变量的存储地址传送给形参。因此，在被调用过程中，对形参的任何操作都是对相应的实参进行的，实参的值会随着形参值的改变而改变。当过程调用结束后，实参变量的值就会发生相应的变化。所以，用这种传送方法可以将过程处理的结果带回到调用程序。

【例4-2】 编写一个用于交换两个数的过程chngXY。

在程序代码中经常要将两个数据进行交换，交换的算法非常简单，只要借助于一个第三单元即可。算法既可以用函数来实现，也可以用过程来实现，用函数实现的代码如下：

```
Public Function chngXY(intX As Integer, intY As Integer)
    Dim intT As Integer
    intT = intX: intX = intY: intY = intT
End Function
```

在Vbxx工程中新建一个标准模块（名称为funChng_4_2），在代码窗口中输入以上代码；然后新建一个名为frmExam_4_2的窗体，在窗体的Click事件中输入以下命令即可调用该函数：

```
inta = 3 : intb = 4
Print inta, intb                '显示交换前的数据
chngXY inta, intb               '不需返回值的调用
Print inta, intb                '显示交换后的数据
```

采用按地址传送方式时，对应的实参不能是表达式或常数。

由于函数过程通过函数名只能返回一个值，如需返回多个值，可以使用按地址传送的方法。

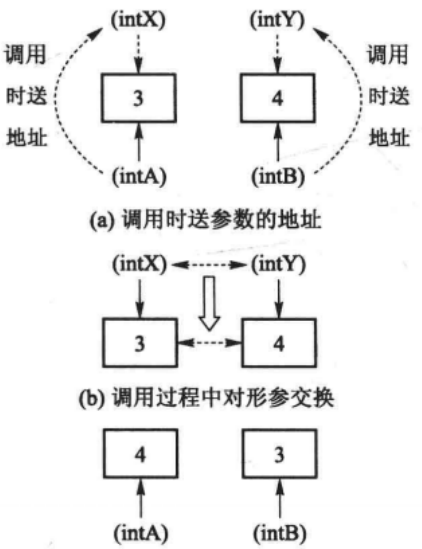
4.3.2 按值传送

按值传送通过在过程定义时在形参前加上 ByVal 来标识。按值传送的形参与实参是这样结合的:调用过程时,系统将实参的值复制给形参,这样形参与实参的联系就断开了。被调用过程是在形参自己的存储单元中进行的,当过程调用结束时,这些形参所占用的存储单元也同时被释放。所以过程中对形参的任何操作都不会影响到实参的内容,也就是说,当返回调用程序时,实参变量将保持调用前的值不变。

在例 4-2 中,如将过程定义变成如下按值传送的方式,其他均不变,再运行程序会发现数据根本没有交换。

```
Public Function chngXY(ByVal intX As Integer, ByVal intY As Integer)
```

两种传送方式可分别用图 4-5 和图 4-6 来表示。

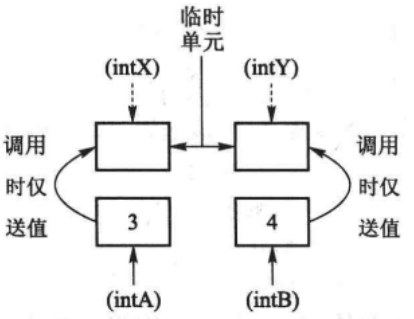


(a) 调用时送参数的地址

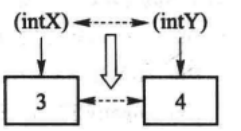
(b) 调用过程中对形参交换

(c) 调用后实参也交换了

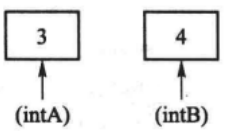
图 4-5 按地址传送参数



(a) 调用时将实参的值送形参的临时单元



(b) 调用过程中对形参交换



(c) 调用返回后实参没有交换

图 4-6 按值传送参数

*4.3.3 数组参数

当需要传送批量数据时,使用数组作为参数是最好的选择(如需处理一批学生的成绩)。但使用数组作为参数传递与使用普通的参数稍有不同。前面已经提过,数组分为固定数组与动态数组两种。在过程定义中将数组作为形参时,必须略去数组的长度,即将数组声明成一个动态数组。这是因为过程在调用前无法预知数组的长度。在调用时可以使用任意长度的数组作为输入值,在调用语句中只需写出要传递的数组名即可,不必包括下标及括号。

【例 4-3】 编写一个过程。对某个班学生的一门课程成绩进行排序,学生人数、学号数据和成绩数据作为参数传递,排序后的数据放在原数组中。

在Vbxx工程中新建一个名为frmExam_4_3的窗体,打开窗体代码窗口,添加一个通用过程(subArgarr)用于对数据进行排序,代码如下:

```
'学号及成绩数组按地址传送,学生数按值传送
Private Sub subArgarr(arrCj( ) As Single,  _
        arrXh( ) As Integer,  _
        ByVal intN As Integer)
    Dim i,j As Integer
    Dim sngTc,intTx As Single
    For i = 1 To intN
        For j = i To intN
            If arrCj(i) < arrCj(j) Then
            sngTc = arrCj(i)
            arrCj(i) = arrCj(j)
            arrCj(j) = sngTc
            intTx = arrXh(i)
            arrXh(i) = arrXh(j)
            arrXh(j) = intTx
        End If
    Next j
Next i
End Sub
```

在窗体的Click事件中输入如下的调用代码(假设有5名学生):

```
Private Sub Form_Click( )
    Dim i,j,intMax,intXh(5) As Integer
    Dim strMsg As String
    Dim sngSq(5) As Single
    '输入原始数据
    For i = 1 To 5
        strMsg = "请输入第" & Str(i) & "个学生的成绩:"
        sngSq(i) = Val(InputBox(j,"数据"))          '将输入的数据转换为数值型
        intXh(i) = i
    Next
    subArgarr sngSq,intXh,5                         '调用排序过程
    For i = 1 To 5                                  '输出排序结果
        Print "第" & Str(i) & "名:";
        Print intXh(i), sngSq(i)
    Next
End Sub
```

在过程的参数传递中,除了可以使用变量作为参数以外,作为面向对象的程序设计语言,VB还可以使用对象作为参数来传递。对象可以是窗体,也可以是控件,当使用对象作为参数来传递时,过程定义中的参数定义只能是按地址传递,不能定义成按值传递。使用对象参数,可以有效提高程序的运行效率,也可以简化程序,特别适用于需要对窗体或控件进行动态生成与设置的程序。有关对象参数的传递请读者参见相关资料,本书不再述及。

4.4 变量与过程的作用域

4.4.1 作用域概述

一个完整的应用程序是由保存在窗体文件和标准模块文件中的若干个过程构成的，而在一个过程中又要使用不同的变量。变量和过程在定义后不是在任何地方都可以被引用的，每个变量或过程都有它自己的作用范围。所谓作用域是指在程序中能够访问过程或变量的代码范围。

过程的作用域比较简单，它只有两种情况：窗体/模块级和公用级。

窗体/模块级的过程也称为局部过程，是指在定义过程时加上 Private 关键字。这种过程只能被本窗体或本模块内的其他过程调用。所有的事件过程都是局部的。

公用级过程是指在定义过程时加上 Public 关键字的过程，公用过程是默认过程。这种过程可以被应用程序内的所有窗体或模块内的其他过程所调用，但调用公用过程要考虑两种情况：

① 如该过程是在窗体模块中定义的，外部过程调用时必须加上该过程所在的窗体名称。

② 如该过程是在标准模块中定义的，所有的外部过程均可调用，但过程名必须是唯一的，否则也必须加上标准模块名。

对于变量而言，其作用域有 3 种：过程级变量、窗体/模块级变量和全局变量。

4.4.2 过程级变量

在过程内部用 Dim 语句声明的变量，称为过程级变量，过程级变量只有在该过程执行时才存在。过程一结束，该变量也就消失了。此外，过程中的变量对过程来说是局部的，也就是说，用户无法在一个过程中去访问另一个过程中的变量，因此这种变量也称为局部变量。使用局部变量的最大好处在于，可以在不同的过程中使用相同的变量名，不必担心会由于变量名的相同而产生冲突，而且也有利于程序的调试。一般情况下，要尽可能地使用过程级变量。

4.4.3 窗体/模块级变量

在一个窗体或模块中会包含若干个过程，这些过程可能要使用某些相同的变量。这些变量的作用域已经不局限于过程内部，它已经将范围扩大到整个窗体或模块中了。在窗体或模块中，所有过程都可以引用的变量称为窗体/模块级变量。

窗体/模块级变量的声明方法与过程级变量的声明方法不同，它一般在窗体/模块代码窗口的“通用说明”段使用 Private 或 Dim 关键字来声明。

4.4.4 全局变量

在应用程序中总存在这样一些变量，它们的作用范围是针对整个应用程序的，而不是仅局限于过程或模块的。例如，用户名、权限和系统日期等。全局变量是指可用于应用程序的所有过程的变量，它的值只有在应用程序执行结束后才会消失。全局变量也称为公用变量。公用变量的声明必须在窗体/模块的通用声明段中使用 Public 关键字。通常的做法是为应用程序添加一个标准模块，然后将所有的全局变量都放在其通用声明部分。因为如果将一个全局变量放在某一个

窗体的声明部分,这个变量只能在本窗体中直接引用,其他过程引用时必须加上窗体名称。

全局变量的好处是显而易见的,一经声明到处都可以使用。但在应用程序中并不提倡使用大量的全局变量,这是因为一方面公用变量声明后直到程序运行结束才能释放,因此内存被严重浪费;另一方面由于任何过程都可以修改公用变量的值,很容易造成程序运行的错误甚至根本不能运行。所以,公用变量尽量少用,甚至不用。

【例4-4】 变量的作用域。

新建两个窗体,分别命名为:frmExam_4_4 和 frmExam_4_41,再添加一个标准模块(main1)。

在标准模块文件中输入以下代码:

```
Option Explicit
Public strGt As String                          '定义全局变量
Public Sub main( )
strGt = "strGt: 标准模块中定义的全局变量"
Debug. Print strGt                              '输出全局变量 strGt
frmExam_4_4. subVar1                            '调用 frmExam_4_4 窗体中的公用过程 subVar1
Debug. Print strGt                              '公用变量已经改变
End Sub
```

窗体 frmExam_4_4 中的代码如下:

```
Option Explicit
Public strGt1 As String                         '定义全局变量 strGt1
Private strMt1 As String                        '定义模块级变量 strMt1
Public Sub subVar1( )
Dim strT As String                              '定义过程级变量 strT
strT = "strT: 过程 subVar1 中的局部变量"
strMt1 = "strMt1: 窗体 frmExam_4_4 中的窗体变量"
strGt1 = "strGt1: 窗体 frmExam_4_4 中的全局变量"
strGt = "strGt: 对公用变量 strGt 进行了修改"    '对全局变量 strGt 进行修改
Debug. Print strT
Debug. Print strGt
Debug. Print strMt1
Debug. Print strGt1
frmExam_4_41. subVar2                           '调用窗体 frmExam_4_41 中的公用过程 subVar2
End Sub
```

窗体 frmExam_4_41 的代码如下:

```
Option Explicit
Private strMt1 As String                        '定义模块级变量,变量名仍为 strMt1
Public Sub subVar2( )
Dim strT As String                              '定义过程级变量,变量名仍为 strT
strT = "strT: 过程 subVar2 中的局部变量"
strMt1 = "strMt1: 窗体 frmExam_4_41 中的窗体变量"
Debug. Print strT
```

```
Debug. Print strMt1
End Sub
```

以上代码中,要特别注意 strT、strMt1 和 strGt 这 3 个变量,它们分别是过程级、模块级和全局级的。strGt 由于是全局级的,任何过程均可对其修改,而 strT 和 strMt1 在两个过程和模块中都存在,但它们相互独立、互不干扰,输出时都是本模块或本过程中的值。

4.5 程序的调试与排错

从程序代码的角度看,一套完整的应用程序通常包含若干个过程、函数。对开发人员来说,程序代码的编写只解决了问题的一部分。因为在程序编写的过程中,不可避免地存在着各种各样的错误,为了改正这些错误,任何形式的程序代码在编写过程中都需要经过调试。程序调试的过程实际上是发现错误、修改错误的过程。

4.5.1 程序错误类型

VB 程序的错误大体上可以分为 3 类:语法错误、运行错误和逻辑错误。

1. 语法错误

语法错误是程序中最容易出现也是最好修改的错误。主要是由于键盘输入错误、命令格式不对或粗心大意等原因造成的。如关键字错,表达式不对,使用了全角的标点符号,函数或过程未定义,语句对不匹配,等等。一般的语法错误在一行代码输入完成后 VB 就会及时检测出来,用红色标出出错的代码行,并显示一个消息框告知产生错误的原因。但也有一些错误要在编译或解释执行时才能检测出来,如语句对匹配错误。

2. 运行错误

运行错误是指程序在输入或编译时没有出现错误,但在运行代码时产生了错误而导致了运行的中断。这类错误往往是由于在指令代码中出现了一些非法的操作而引起的。例如试图打开一个不存在的文件、被 0 除、磁盘空间不足、使用了“要求变量声明”但变量未定义等。当出现运行错误时,屏幕也会出现消息框,如图 4-7 所示即为打开一个不存在的文件而出现的消息框。消息框中一般有 4 个按钮,其作用非常明显。单击“调试”按钮时,程序进入中断模式,光标停留在引起错误的那一行代码上,此时可以修改错误的代码。

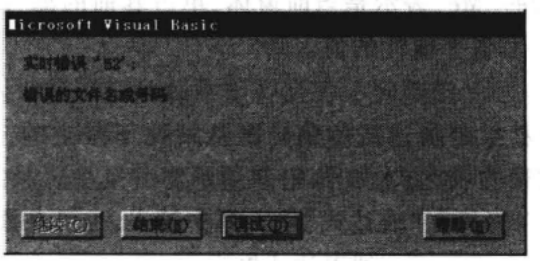


图 4-7 实时错误消息框

3. 逻辑错误

逻辑错误实际上是算法上的错误,这种错误 VB 在编译和运行时是无法发现的,但程序运行

结束后所得到的却是错误的或并非预期的结果。这类错误最麻烦,也最难改正,一般只能通过仔细检查程序代码并借助于VB的调试工具来判断才能排除。

4.5.2 调试与排错

程序的调试过程就是查找那些使程序不能正确运行的错误的过程,这些错误一般都是逻辑错误。利用VB所提供的调试工具可以帮助用户处理这些错误,同时还能观察代码的运行状态。前文已述及,VB的工作模式有3种:设计模式、运行模式和中断模式。程序的调试一般在中断模式下进行,主要使用调试菜单或调试工具栏来进行。

1. 中断模式的进入与退出

进入中断模式的方法有以下几种:

- 直接进入中断模式。首先打开需要调试的程序代码,然后使用单步调试方式,这是最常用的一种方式。
- 程序在运行时发生了运行错误而进入中断模式。
- 用户在程序中设置了断点,当程序运行到断点时也进入中断模式。
- 程序在运行中,用户发出了中断指令(按组合键Ctrl + Break或单击"中断"按钮)亦可进入中断模式。

无论哪种方式进入中断模式后,都可以使用VB提供的调试窗口来查看各个变量及属性的当前值,从而检查程序的执行是否正常。

退出中断模式有两种方法:一是程序运行结束即退出;另一种是强行结束,即单击"结束"按钮。

2. 调试窗口

VB提供了3个窗口用于调试,它们是:立即窗口、本地窗口和监视窗口。为了便于说明,现将例4-7的程序代码装入到一个窗体的Click事件中,并单击调试工具栏上的"逐语句"按钮即可进入中断模式。进入中断模式后,再使用调试工具栏上相应的按钮打开这3个窗口。

(1) 本地窗口

本地窗口显示当前过程中所有变量的值。随着程序的执行,各个变量的值当然也会发生变化,这种变化会及时地反映到该窗口中。当程序执行时从一个过程切换到另一个过程时,该窗口的内容也会发生变化。图4-8即为执行例4-7过程时的本地窗口。

图4-8中最上面一行的"Me"表示是当前窗体,单击其前的加号(+)可打开该窗体及其各个控件对象的属性树,以了解各个属性的值。

(2) 监视窗口

监视窗口主要用来查看指定表达式的值。默认情况下是空的,可以在该窗口中右击选择"添加监视"来添加需要监视的表达式,如图4-9所示。

监视表达式的添加可以直接在"表达式"输入框中输入,也可在添加监视前先在代码中选定需监视的表达式,这样它就会自动填入到表达式框中。

"上下文"选项组用于选定监视表达式所在的位置。"监视类型"选项组中的3个选项用来控制对表达式的监视方式。当有关的选择结束后,单击"确定"按钮即可回到图4-10所示的监视窗口。在程序运行的过程中,监视窗口的内容也在改变,由此也可判断程序的运行是否正常。

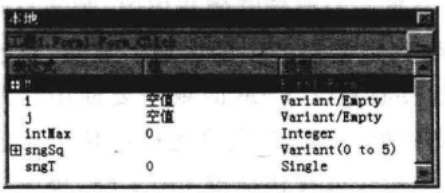


图 4-8 本地窗口

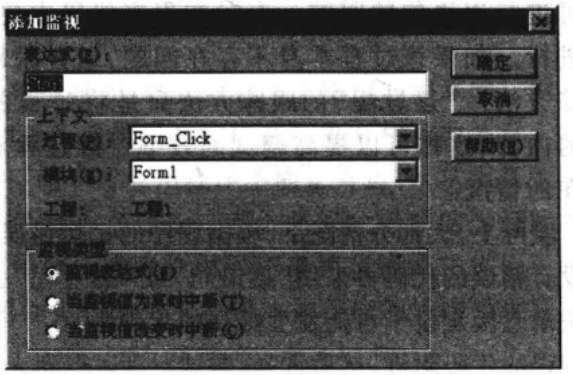


图 4-9 “添加监视”对话框

在监视窗口中随时可以添加其他的监视表达式,也可利用快捷菜单来编辑监视或删除监视。

(3) 立即窗口

立即窗口是一个交互窗口,用于在调试时执行代码或查询某个变量的值。在进入中断模式时默认显示立即窗口。在立即窗口中可以进行下列操作:

① 输出到立即窗口。进入中断模式后,可以在立即窗口用 Print 命令(或用“?”命令)来检查或验证表达式的值。在程序代码中可以使用 Debug. print 方法将变量或表达式的值输出到立即窗口中。如在图 4-11 中即为实时地检查 str(i)的值。

② 重新设置或修改变量与属性的值。在中断模式下可以利用立即窗口人为地修改某个变量或属性的值,以观察其对程序运行的影响。

③ 测试过程。在立即窗口中可以通过指定的参数来直接调用过程,以测试过程的正确性。

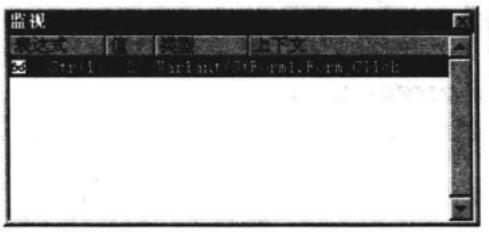


图 4-10 监视窗口

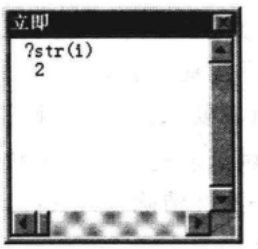


图 4-11 立即窗口

3. 跟踪程序的运行

一个不存在语法错误与运行错误的程序运行后得不到预期的结果,说明程序肯定存在着逻辑错误。此时应仔细地检查程序代码,如果根据错误的运行结果就能判断出错误所在,则直接修改程序即可。但更常见的情况是只知道产生错误的大致区域,这样就需要使用 VB 的跟踪工具来跟踪程序的运行过程。

跟踪的方法有 4 种:逐语句、逐过程、运行到光标处和设置断点。

“逐语句”就是一条语句一条语句地执行代码,通过查看应用程序的窗体和相关的调试窗口来判断该语句的执行是否正确(主要是观察变量和表达式的值是否正确,窗体上显示是否正常等)。单击调试工具栏上的“逐语句”、选择〖调试〗→〖逐语句〗菜单命令或按 F8 键

均可进行逐语句的跟踪。有时在程序代码中要调用过程，这种跟踪也会深入到被调用过程中进行。如果该过程本身不存在问题，这种跟踪就没有必要了，那就可以使用“逐过程”的跟踪方法。“逐过程”的跟踪方法只是将过程的调用作为一条语句来执行，而不深入到过程的内部，这样可以提高调试的效率。逐过程和逐语句的调试方法一般适用于不知道错误所在的情况。

实际上，大部分情况下，根据程序的逻辑错误可以大致判定错误产生的区域。显然，对于那些没有错误的代码再采用“逐语句”或“逐过程”去跟踪就没有意义了，此时就需要使用后两种方法来直接定位于可能发生错误的代码附近。“运行到光标处”就是首先将光标定位于可能产生错误的代码附近，然后选择〖调试〗→〖运行到光标处〗菜单命令或按快捷键 Ctrl + F8，程序立即从第一条语句开始运行直到光标所在的代码行，从而进入中断模式。进入中断模式后再采用“逐语句”或“逐过程”的方法来进行跟踪以确定错误位置。

4. 断点的设置

对于大程序或循环程序而言，采用单步运行效率不高。设置一定的断点是快速跟踪程序的行之有效的方法。“断点”是通知 VB 挂起程序的标记，当程序执行到断点处时即暂停运行而进入中断模式，从而可以观察程序的运行状态。断点通常安排在程序代码中能反映程序执行状态的部位。如程序运行的错误可能与某段循环程序有关时，就可以在循环体内部安排一个断点，循环每执行一次程序即中断一次，可以检查每次循环的循环变量及其他变量的取值情况，从而确定出错的原因。

在程序中设置断点非常简单。在代码窗口中，将光标移向想作为断点的代码行，然后用以下任一种方法均能设置断点：按 F9 键、选择〖调试〗→〖切换断点〗菜单命令或单击调试工具栏上的“切换断点”按钮。在同一段程序代码中可以设置多个断点。要取消断点用同样的方法即可，也可以选择〖调试〗→〖清除所有断点〗菜单命令将所有的断点一次性清除。图 4-12 所示的代码中设置了两个断点。



图 4-12　断点的设置

☞ 单击指定代码行行首的灰色条也可设置或清除断点。

❗ 要将断点设在有效代码上，否则断点将不会起作用。

断点设好后即可运行程序。程序首先在第一个断点处进入中断模式，此时，通过应用程序的窗体界面和调试窗口可以检查程序的状态。如果程序运行是正常的，再通过“逐语句”或“逐过程”来调试以下的代码，直到找出错误的原因。错误的原因一旦查到，修改就变得非常简单了。但要注意的是，对代码的修改一般不要在中断模式下进行，而要在设计模式。从中断模式退到设计模式的方法是直接单击“调试”工具栏上的“结束”按钮。

总之，程序员只有具备严谨认真的工作态度、耐心细致的工作作风并能灵活运用各种调试工具和方法，才能快速纠正程序中的各种错误。

*4.5.3 出错处理

VB 程序可能出现的 3 类错误中,运行错误是最难控制的,而且是无法彻底消除的。因为运行错误的产生总是与用户的使用情况、计算机的运行环境以及文件系统等有相当的关系。如文件的误删除、网络故障和磁盘没准备好等,都有可能导致应用程序不能正常运行,从而产生运行错误。一旦发生这样的错误而且又没有容错处理的话,Windows 系统将终止程序的运行并将控制权归还给 Windows 系统。为此就需要在程序中加入错误处理程序来捕获错误并执行正确的操作。这就要求程序员在开发程序时,在任何可能出错的地方都要添加相应的错误处理程序以防患于未然。

VB 中的错误处理过程大致分为 3 个步骤。

(1) 设置错误陷阱

设置错误陷阱就要用到 On Error 语句。当错误发生时,系统将激活错误陷阱并将程序的执行定向到 On Error 语句指定的错误处理程序中。

On Error 语句有以下 3 种形式。

- On Error GoTo line:启动错误处理程序,该错误处理程序从 line 参数中指定的行标签或行号开始。如果发生一个运行时错误,则程序会跳到 line,然后激活错误处理程序。指定的 line 必须与它所在的 On Error 语句在同一个过程中,否则会发生编译错误。
- On Error Resume Next:当一个运行发生错误时,程序转到发生错误的语句之后的语句继续运行。这种格式一般用在访问一个对象的时候。
- On Error GoTo 0:禁止当前过程中任何已启动的错误处理程序。



一个错误处理程序不是 Sub 过程或 Function 过程。它只是一段用行标签或行号标记的代码。

(2) 错误的处理

当错误被 On Error 语句捕获到后,Err 对象中就会包含有运行时错误的相关信息,该信息的 Number 属性指出了对应的错误号。错误被捕获到后,如何处理错误就是错误处理程序的任务了。错误处理程序是包含在通用过程或事件过程中的用行标号标记的一段代码,它专门用来处理相关的错误,而且能根据可预知的错误类型来决定采取何种措施。显然,程序在正常运行时,该代码是不会执行的。



各种错误的 Number 值请查阅联机帮助中的“可捕获的错误”为关键字的索引。

【例 4-5】 错误处理示例:以下代码演示了如何捕获“数据类型不匹配”的错误及处理方法。

```
Private Sub Form_Click()
  Dim i As Integer
  Dim intS As String
  On Error GoTo Errhandle          '当出现错误时捕获,并转到 Errhandle 行的错误处理代码
  i = 100: intS = "abcd"
  Print intS / i                   '人为地制造一个“数据类型不匹配”的错误
  Errhandle:
```

```
    If Err.Number = 13 Then        '"数据类型不匹配"错误号为13
        MsgBox ("数据类型不匹配", vbCritical, "错误")          '显示错误信息
        Exit Sub                                              '退出该过程
    End If
End Sub
```

(3) 退出错误处理程序

错误处理结束后必须能退出错误处理程序,并恢复程序的运行。实现这一功能要用 Resume 语句。

Resume 语句也有 3 种形式。

- Resume:返回到出现错误的语句重新执行。如果错误发生以后能够立即消除,才能使用这种返回方式,否则仍然会发生同样的错误。
- Resume Next:返回到发生错误的下一行继续执行,这种返回方式相当于忽略了发生错误的语句。如果所发生的错误不会对程序的运行产生致命的影响,才能使用这种方式。
- Resume line:返回到 line 指定的行继续运行并忽略错误行。

因此,究竟采用何种方式返回要根据错误的性质来决定。一般情况下,可以立即消除的错误(如数据输入类型不符、驱动器未准备好或打印机未准备好等)可以使用第一种方法,而对于一些未知的错误则可以使用后两种方法。

有关错误处理的详细内容,限于篇幅,本书不再展开,感兴趣的读者可参考有关资料。

习 题 4

1. VB 中有哪几类过程? 它们之间有何异同?
2. 何为形参? 何为实参? 进行参数传递时应注意哪些问题?
3. 在什么情况下才需要使用公有过程?
4. 过程与变量的作用域是如何规定的?
5. 事件过程与方法有何区别?
6. 地址传递与值传递有什么区别? 分别应用在什么情况下?
7. 编写一个过程用于对 20 个学生的成绩进行排序,过程参数用两个数组:一个存放学生成绩,另一个存放学生的学号。排序后的结果仍然通过这两个数组返回。
8. 如下程序,试分析程序的运行结果(请注意变量的作用域)。

```
Dim a%, b%, c%
Private Sub cmdEnd_Click()
    a = 2: b = 4: c = 6
    Call p1(a, b)
    Print "a="; a; "b="; b; "c="; c
    Call p2(a, b)
    Print "a="; a; "b="; b; "c="; c
End Sub

Public Sub p1(x%, y%)
```

```
    Dim c%
    x = 2 * x: y = y + 2: c = x + y
End Sub

Public Sub p2(x%, ByVal y%)
    Dim c%
    x = 2 * x: y = y + 2: c = x + y
End Sub
```

9. 编写实现十进制与其他进制转换的函数过程。十进制数使用 InputBox 函数输入,调用函数后输出相应的二进制数、十六进制数和八进制数。

10. 应用程序会产生哪些错误? 如何调试并修改这些错误?

11. VB 提供了哪些程序调试工具? 如何使用?

12. 程序调试题:过程 DelStr(s1,s2)函数的功能是将字符串 s1 中出现的 s2 字符串删除,结果仍然存放在 s1 中。为调用该过程,新建一个窗体,在窗体上放置 3 个文本框(Text1,Text2,Text3),分别存放 s1、s2 和删除后的 s1 值。另放置一个命令按钮用于实现过程的调用。相关的程序代码如下:

```
Public Sub DelStr(s1 As String, s2 String)
    Dim i As Integer
    i = InStr(s1, s2)
    ls2 = Len(s1)
    Do While i > 0
        ls1 = Len(s2)
        s1 = Left(s1, i) + Right(s1, ls1 - (i + ls2))
        i = InSr(s1, s2)
    Loop
End Sub
Private Sub Command1_Click()
    Dim ss1 As String
    ss1 = Text1. Value
    Call DelStr(ss1, Text2. Text)
End Sub
```

要求:

(1) 创建该窗体,原样输入以上的代码并修改代码中的语法错误。

(2) 试运行该窗体,修改代码中的运行错误。

(3) 利用调试工具修正代码中的逻辑错误。

第5章　VB中的窗体与常用控件

学习目标

熟练掌握VB中常用控件(单选按钮与复选框、列表框与组合框等)的使用方法,包括控件的常用属性、事件及事件过程的代码设计、控件的应用场合等。熟悉其他控件(滚动条、定时器和框架等)的应用。熟练掌握窗体的使用与控制,能根据一般应用程序的需要设计符合要求的程序界面。

在第2章已经学习了标签、命令按钮和文本框等3个典型控件的基本使用方法,但只通过这3种控件尚不足以设计完整的应用程序。本章将介绍其他一些常用控件和窗体的使用,以使读者对VB控件的使用有一个较为全面的了解,能够将窗体和控件等界面元素有机地结合起来,从而掌握搭建一个完整的、易学易用的应用程序。

【案例三】 学生成绩通用查询条件生成器。

在一般的信息管理系统软件设计中,经常要进行各种数据的查询。查询的方式通常都是通过SQL命令结合数据库来进行的。在查询中往往要指定查询的条件,如"姓名="张三""、"课程="计算机网络""等。这些条件实际上就是在第3章中介绍的各种关系表达式。关系表达式可以在程序中事先指定(这种查询实际上是固定查询),但使用更多的是动态查询,即查询条件由用户根据情况指定。本案例就是利用VB制作一个条件表达式的生成窗体(见图5-1),利用该窗体来执行动态的通用成绩查询。

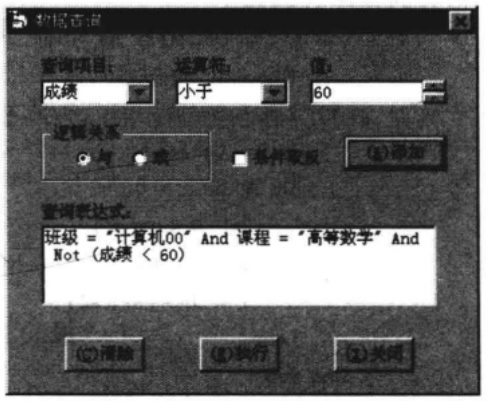


图5-1　通用查询条件生成器

5.1　控件应用基础

VB的控件尽管种类繁多,使用设置也不尽相同,但仍然还有许多相同或相似的地方。在第2章已经介绍了控件的常用属性。除了常用属性以外,控件还有一些常用的事件及其他一些共性问题,理解并掌握控件中的这些共性问题对程序的设计将起到事半功倍的效果。

5.1.1　常用事件

VB中的控件能接收各种各样的事件,在这些事件中有几个事件是绝大部分控件都具有的,在程序设计中使用也非常普遍。

1. Click事件

Click事件是控件中使用最为普遍也是最为重要的一个事件。VB中的窗体与绝大部分控件都能响应该事件。当下列情况发生时会引发该事件,同时执行Click事件中的代码:

- 单击一个控件。

- 当按钮、单选钮、检查框具有焦点时，按下 Space 键。
- 通过箭头键或鼠标在组合框、列表框中选择项目时。
- 在窗体中存在 Default 属性为 True 的按钮且按 Enter 键时。
- 在窗体中存在 Cancel 属性为 True 的按钮且按 Esc 键时。
- 按下某一个控件的访问键时。

要注意的是，当 Click 事件发生时，还会伴随着其他事件的发生。它们是：MouseDown（鼠标按下）事件和 MouseUp（鼠标松开）事件。其发生的顺序一般为：MouseDown、Click、MouseUp。一般情况下对 MouseUp 和 MouseDown 这两个事件并不编写代码，因此也就不会响应了。

2. DblClick 事件

DblClick 事件也称为双击事件，它是通过双击鼠标触发的。在 Windows 应用软件的对话框中，有许多场合可以使用鼠标双击来完成预定的功能。如在文件打开对话框中双击某个文件夹表示打开该文件夹，在列表框中双击某个项目表示移动该项目等。这些功能正是通过编写某一控件的 DblClick 事件代码得以实现的。

VB 中支持双击事件的控件相当多（但命令按钮不支持），它主要应用于列表框、组合框、文件列表框、数据绑定列表框等。双击事件只能通过鼠标对控件或项目的双击而触发。

事实上，从应用软件的使用上可以看出，双击操作只是一个辅助操作，它实际上相当于选择某个项目后（如打开文件对话框中的某个文件）再单击某个按钮（如打开）。因此，在编写某个控件的 DblClick 事件代码时，一般只要显式调用某个命令按钮的 Click 事件过程即可。例如：

```
cmdEdit. value = true
```

3. GotFocus/LostFocus 事件

在第 2 章中讲述了“焦点”的概念，只有当对象具有焦点时，才能接受键盘的输入或响应鼠标。焦点只有一个，一个对象得到焦点也就意味着其他对象要失去焦点。按 Tab 键或单击对象时，会引起焦点的移动。对象得到焦点时将触发 GotFocus 事件，而对象失去焦点时将触发 LostFocus 事件。

只有对象的 Enabled 和 Visible 两个属性均为 True 时，才能获得焦点。

通过给窗体上某些控件附加 GotFocus 事件过程，可以显示简要说明或状态信息，给用户提供必要的指导。

LostFocus 事件主要是用来对数据更新进行验证和确认，例如，当文本框的内容输入完成后，焦点移离文本框时 LostFocus 事件代码将执行，这样就可以检查所输入的内容是否正确。这类事件过程的另一种用途与在 GotFocus 事件过程中的应用类似，也可以提供某些说明信息或隐藏、显示其他对象或者使它们有效或无效。

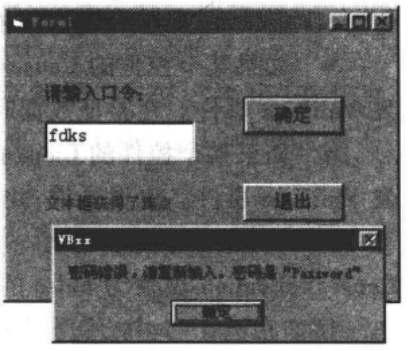


图 5-2 例 5-1 运行情况

【例 5-1】 GotFocus/LostFocus 事件应用举例。

新建一个名为 frmExam_5_1 的窗体，在窗体上放置两个标签（lblPass，lblProm）、一个文本框（txtSrxx）与两个命令按钮（cmdOk，cmdExit），各控件的其他属性设置如图 5-2 所示。

自本章开始，对窗体与控件上要设置的常用属性（如 Caption、Name 等）将不再特别说明。

有关的事件过程如下：

```
Private Sub cmdOk_GotFocus( )                          '"确定"按钮获得焦点事件
    lblProm. Caption = "'确定'按钮获得了焦点"          '设置标签的 Caption 属性
End Sub
Private Sub cmdExit_Click( )                           '"退出"按钮的单击事件
    Unload Me
End Sub
```

```
Private Sub cmdExit_GotFocus( )
    lblProm. Caption = "'退出'按钮获得了焦点"
End Sub
```

```
'文本框获得焦点事件
Private Sub txtSrxx_GotFocus( )
    lblProm. Caption = "文本框获得了焦点"
End Sub
```

```
'文本框失去焦点事件
Private Sub txtSrxx_LostFocus( )
    If txtSrxx. Text < > "Password" Then
        MsgBox ("密码错误,请重新输入。密码是'Password'")
        txtSrxx. SetFocus                              '重新获得焦点
    End If
End Sub
```

该程序的运行情况如图5-2所示，请读者自行在计算机上演示该程序。

4. ValiDate事件

几乎所有的控件都具有ValiDate事件，ValiDate事件是在控件的焦点转移到其他控件之前发生，即它是在控件失去焦点之前发生，而LostFocus事件是在控件失去焦点之后发生。因此，ValiDate事件更适合数据输入的验证。

一般的按钮、文本框和列表框等控件都有一个CausesValidation属性，只有True和False两个值，它用来控制当焦点发生转移时是否引发ValiDate事件。因此，ValiDate事件应与控件的CausesValidation属性联合使用才能发挥作用。

- 如果某个控件的CausesValidation属性设置为True，则当焦点转移到其他控件时即会触发ValiDate事件。在验证数据时，将不能再响应其他任何事件。
- *如果某个控件的CausesValidation属性设置为False，则当焦点转移到该控件时就不会触发ValiDate事件。*也就是说，此时不再验证数据了。
- ValiDate事件中还有一个Cancel参数，当其值为True时将保持焦点不变，但它不影响CausesValidation属性为False的控件获得焦点。

例如在例5-1中，使用了LostFocus事件来验证数据密码输入的合法性，现在将其改为用ValiDate事件来验证，并且要求：

① 如果密码输入不对，保持焦点不变。

② 无论密码输入正确与否，单击"退出"按钮总能退出。

实际上,要实现以上功能并不复杂,只要首先将“退出”按钮的 CausesValidation 属性改为 False,然后删除原来的 LostFocus 事件代码,再编写文本框的 ValiDate 事件代码如下:

```
Private Sub txtSrxx_Validate(Cancel As Boolean)
    If txtSrxx.Text < > "Password" Then
        MsgBox ("密码错误,请重新输入。密码是'Password'")
        Cancel = True                                    '保持焦点
    End If
End Sub
```

程序的运行情况请读者自行验证。

5. KeyPress 事件

KeyPress 事件是在用户按动键盘上任何一个可打印的字符时发生的,只有能接收键盘输入的控件才有 KeyPress 事件。KeyPress 事件过程用得最多的是文本框和组合框,它可以截取 TextBox 或 ComboBox 控件所输入的击键,而且可立即测试击键的有效性或在字符输入时对其进行一定的格式处理。

KeyPress 事件还有一个参数 KeyAscii,用来返回击键的 ASCII 码,利用这个参数可以对所输入的字符进行格式转换。例如,用下面的语句即可实现将输入的文本全部转换为大写。

```
Char = Chr(KeyAscii)
KeyAscii = Asc(UCase(Char))                    'UCase()为大写转换函数
```

KeyPress 事件并不能识别功能键(F1 ~ F12)和编辑键(如 Insert、Home 等)的击键动作,要捕获这些按键可以采用 KeyDown 和 KeyUp 事件过程。

5.1.2 Tab 键顺序

在 Windows 应用软件的使用中,要使控件获得焦点可以通过鼠标与键盘两种方法。利用键盘获得焦点,也可以通过两种途径来实现:为控件设置访问键和利用 Tab 键。所谓“利用 Tab 键”是指在按动 Tab 键时,焦点在控件之间的转移顺序。

在 VB 中,除 Menu、Timer 和 Data 等以外的控件都有一个属性值,称为 TabIndex,即 Tab 键顺序号,它决定了控件在 Tab 键中的顺序。默认情况下,Tab 键顺序号就是在窗体上添加控件的次序,画到窗体上的第一个控件其 TabIndex 值为 0,第二个为 1,以此类推。如在例 5-1 中的 5 个控件中,按照绘制次序其 TabIndex 分别为:lblProm-0、txtSrxx-1、cmdOk-2、cmdExit-3、lblPass-4。在该程序的运行过程中,当连续按动 Tab 键时,焦点即会在文本框、“确定”按钮和“退出”按钮 3 个控件间循环切换(当然假设文本框中不含有 ValiDate 与 LostFocus 事件过程时)。

对于无效的、不可见的以及不能接受焦点的控件,尽管都具有自己的 TabIndex 属性值,但 Tab 键将跳过这些控件。

显然,只要改变控件的 TabIndex 值即可以改变其在 Tab 键中的顺序。当一个控件的 TabIndex 值改变或某个控件删除后,系统会自动对其他控件的 TabIndex 属性重新编号。改变某个控件的 TabIndex 值既可以在设计时,也可以在运行时通过代码来修改,例如:

```
cmdOk.TabIndex = 0
```

该语句即将确定按钮的Tab键顺序改为第一个。

*5.1.3　控件数组

在VB中除了有静态数组与动态数组以外，还有一种特殊的数组，即控件数组。控件数组是由具有相同名称和类型并具有相同事件过程的一组控件构成。每个控件数组中至少包含有1个元素，最多可有32 767个元素。数组中的下标值即表示控件数组中的元素序号，第一个下标也是0。当建立控件数组时，系统为每一个元素赋予了唯一的索引号，这个索引号放置在控件属性窗口的属性Index中。因此，通过查看某控件的Index值，就可以知道它是控件数组中的第几个元素。

1. 控件数组的应用

为什么要使用控件数组呢？首先，在程序设计时，使用控件数组添加控件所消耗的资源比直接向窗体添加多个相同类型的控件消耗的资源要少。其次，如果希望若干控件共享代码时，控件数组也很有用。例如，在应用软件的窗体上经常会包含若干个命令按钮，如果将这些命令按钮放在一个控件数组中，只要编写一个Click事件过程代码即可由数组中的所有按钮共享。也就是说，无论单击哪个按钮时，都将执行相同的事件过程。

为了区分控件数组中的哪个元素接受了用户的动作，VB会将接受动作的元素的下标值传递给过程，这个下标值正是该元素的Index值。事件过程的代码中只要使用判定结构即可调用相应的程序代码以完成预期的功能。

例如，命令按钮控件数组的Click事件过程的代码结构可能是如下形式：

```
Private Sub cmdGroup_Click(Index As Integer)          'Index为引发该事件的按钮值
    Select Case Index
        Case 0                                        '按第一个按钮时执行的代码
            ...
        Case 1                                        '按第二个按钮时执行的代码
            ...
    End Select
End Sub
```

2. 控件数组的创建

创建控件数组有两种不同的方法：一种是设计时创建，另一种是运行时创建。本书主要介绍第一种方法，第二种方法读者可参考其他资料。

设计时创建控件数组又可分为以下两种情况。

① 通过复制现有的控件来创建控件数组。方法是：先在窗体上绘制出控件数组中的第一个控件并进行必要的设置，然后再复制该控件，最后粘贴。如此反复进行就可以建立控件数组。第一次粘贴时系统会弹出“是否创建控件数组”的消息框，单击“是”按钮即可。用这种方法创建控件数组中的每个控件除了Index（第一个控件的值为0，第二个为1，以此类推）和TabIndex属性外，其他可视类的属性（如Height、Width、ForeColor等）都与第一个控件相同。

② 用已经存在的多个同类控件来创建控件数组。方法是：先确定要作为第一个元素的控件，然后将其他控件的名称值改为与第一个控件相同的名称值，此时系统也会弹出确认的消息框，回答“是”后，控件数组就创建好了。用这种方法创建的控件数组中的控件除了名称、类型与

第一个控件相同外，其余仍保持该控件创建时的值。

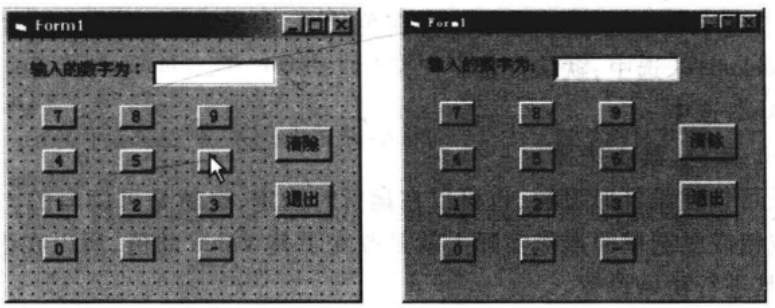
【例 5-2】 模拟数字按键。

要求：在一个窗体上设计一个模拟的数字键盘，将单击的数字显示在文本框中并能清除文本框中的数字。

窗体的设计界面如图 5-3a 所示，本窗体采用了 1 个标签（lblNum）、1 个文本框（txtNum）、2 个按钮控件数组。按钮控件数组 cmdNum 中包含了 12 个数字按钮，按钮控件数组 cmdCom 中包含了 2 个命令按钮（“清除”与“退出”）。所编写的事件过程也非常简单，如以下代码所示，请特别注意 Index 的使用。

```
Private Sub cmdcom_Click(Index As Integer)
    If Index = 0 Then txtNum.Text = ""          '如索引号为 0 表示单击了“清除”按钮
    If Index = 1 Then Unload Me                 '如索引号为 1 表示单击了“退出”按钮
End Sub
```

```
Private Sub cmdNum_Click(Index As Integer)
    txtNum.Text = txtNum.Text + cmdNum(Index).Caption
                                                '将单击的数字连续地显示在文本框中
End Sub
```



(a) 例5-2窗体布局　　(b) 例5-2运行结果

图 5-3　例 5-2 程序及运行结果

5.2　单选按钮与复选框

用户对应用程序的使用通常是通过菜单、快捷菜单及对话框实现的。在使用过程中，有许多情况需要用户做出选择（如性别、特长、课程和班级等），这些选择有时候范围很小（只需回答 Yes/No），而有时候范围很广（可能是长长的列表）。VB 提供了制作这类选择的控件：适用于小范围选择的单选按钮与复选框；适用于大范围选择的列表框与下拉列表框。本节先讨论前者，后者将在下节讨论。

5.2.1　单选按钮与复选框的创建

相信读者在许多 Windows 应用软件中使用过单选按钮与复选框这两种屏幕元素。单选按钮（OptionButton）与复选框（CheckBox）的区别有两点：第一，两者的表现形态不同，单选按钮为圆形而

复选框为方形;第二,在一组单选钮中用户只能选择其中之一,而复选框可以单选也可以多选。

创建单选按钮与复选框的方法与创建一般的控件相同,可以通过双击控件工具箱中的OptionButton或CheckBox控件类来创建,也可以通过单击后在窗体拖动的方法来创建。一般来说,都要将它们放置在相应的控件数组中以便管理。

5.2.2 设置单选按钮与复选框的状态

状态设置主要是指对单选按钮与复选框的各种属性进行设置。对单选按钮与复选框而言,它们也具有诸如Name、Caption、Width、Height、Top、Left、ForeColor、BackColor、Enabled、Visible、TabIndex等属性,对这些属性的设置方法与含义在前几章及本章的5.1节已经进行了阐述。在本书以后的叙述中,为节省篇幅,对常规属性的设置如无特殊情况将不再说明。

1. Value的设置

Value决定了单选按钮或检查框是否被选中,用来表示其状态。

(1) 单选按钮

单选按钮只有两种状态:选中或未选中,如选中,Value值为True;如未选中,Value值则为False,默认为False。

(2) 复选框

复选框除了有选中与未选中外,还有一种灰色状态,即禁止选择,其取值如下。

- 0 - Unchecked:未选中,默认设置。
- 1 - Checked:选中。
- 2 - Grayed:灰色,禁止选择。

Value属性既可以在设计时设置,也可以在运行时确定,也就是可读写的。在程序执行过程中,用户对单选按钮或复选框的操作实际上就是改变了其Value的值,所以在设计时对Value值的设置仅仅是对其进行初始设置。

2. Style的设定

Style(风格)决定了单选按钮与复选框的显示形式,主要是用来改变其视觉效果。它有两种选择。

- 0 - Standard:标准方式,默认值。
- 1 - Graphical:图形方式。当设置成图形方式后,就可以在其上设置位图或图标(Picture、DownPicture和DisabledPicture三种属性)以增强视觉表现力。

5.2.3 单选按钮与复选框的使用

在程序中使用单选按钮与复选框主要要解决两个问题:第一是当用户对单选按钮或复选框做出响应后,能准确判定出响应的位置及其取值;第二是在程序中如何主动地设置单选按钮或复选框的各种状态。

在单选按钮或复选框中最常用的事件是Click事件,显然利用该事件可以很容易地得到控件的取值。如果是控件数组的话还可以得到触发Click事件的控件位置。这样第一个问题就得到了解决。第二个问题也比较简单,只要在适当的事件代码中对控件的可读写属性进行必要的设置就行了。

【例 5-3】 设计一个简单的留言簿,要求能改变输入文字的字体及字形。

设计步骤如下:

① 在 VBxx 中新建一个窗体,取名为"frmExam_5_3"。

② 在窗体上放置 1 个标签、2 个命令按钮(数组)、3 个单选钮(数组)、3 个复选框(数组)和 1 个文本框。各控件的属性设置如表 5-1 所示。

表 5-1 例 5-3 控件主要属性的设置

控件数组	属性名称	取值
单选钮 OptFont(1)	Value	True
命令按钮 cmdGrp	Default	True
文本框	Multiline	True
	ScrollBars	2 - Vertical

设置后窗体布局如图 5-4 所示。

③ 相关的事件过程代码如下:

```
Private Sub chkBIU_Click(Index As Integer)                    '复选框控件数组的单击事件过程
    Select Case Index
      Case 0
        txtLyb.FontBold = Not txtLyb.FontBold                 '改变粗体状态
      Case 1
        txtLyb.FontItalic = Not txtLyb.FontItalic             '改变斜体状态
      Case 2
        txtLyb.FontUnderline = Not txtLyb.FontUnderline       '改变下划线状态
    End Select
End Sub
```

```
Private Sub cmdGrp_Click(Index As Integer)                    '命令按钮数组的单击事件过程
    If Index = 1 Then Unload Me
    If Index = 0 Then                                         '清除文本框,并重置焦点
        txtLyb.Text = ""
        txtLyb.SetFocus
    End If
End Sub
```

```
Private Sub optFont_Click(Index As Integer)                   '单选按钮数组单击事件过程,设置字体
    If Index = 0 Then txtLyb.FontName = "宋体"
    If Index = 1 Then txtLyb.FontName = "黑体"
    If Index = 2 Then txtLyb.FontName = "楷体_GB2312"
End Sub
```

编写以上代码时,要注意 Windows 的字体名称不能写错,标准的字体名称可以从 Font 对话框中获得。

④ 事件代码编写完成后进行必要的调试,运行的情况如图 5-5 所示。

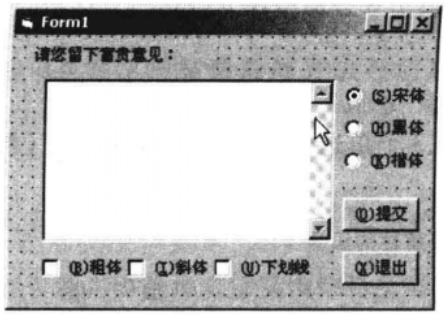


图5-4　留言簿窗体布局

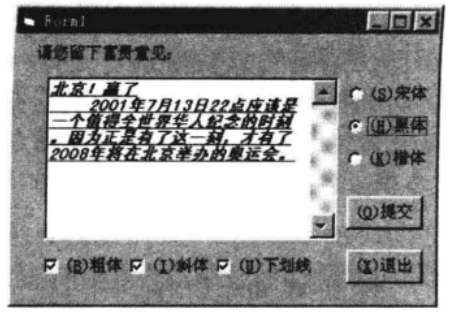


图5-5　留言簿运行情况

5.2.4　框架的使用

单选按钮有一个重要的特点是当选中其中一个时，其他的会自动取消选中。另外，VB总将放置在窗体上的所有单选按钮都作为一组。但有时需要在窗体上建立几组相互独立的单选按钮组以进行不同的功能选择。如上例中字体是一个组，如果想再建立一组用于选择字号的单选按钮组，就必须使用框架了。

框架(Frame)是一种能将窗体上的控件进行分组的特殊控件，在一个框架内部通常设置同一种类型的控件。当框架移动时，其内部控件也会随之移动。使用框架一方面可以将窗体上的控件按功能进行视觉上的区分；另一方面主要是为了建立多个不同的单选按钮组，这样各个单选按钮的操作不会相互影响。

由于框架包含两个层次：内部控件与框架本身，因此在创建时一般应先在窗体上绘制框架，然后在框架中再建立其内部控件。如果想将已有的控件放置到框架中，必须先剪切这些控件，然后再粘贴到框架中。

☞ *在框架中建立控件时，不能采用双击控件类的办法，只能使用单击加拖动的方法。*

框架也具有一些常规的基本属性，如名称、Caption、Enabled和Visible等。设计时主要是设置其Caption属性，即框架上的标题。要注意的是，有些属性的设置不仅影响框架本身，同时对框架内部的控件也有影响。可以设想框架是一个容器，其内部控件将随着该容器一起移动、显示、隐藏或禁止。例如，将框架的Enabled属性设置为False，则在程序运行时框架的标题将变成灰色，内部的所有控件都不能响应任何事件。同样将框架的Visible属性设置为False，则程序运行时框架及其内部控件均被隐藏。

框架本身也能响应Click及DblClick事件，但在实际的应用程序中极少对这些事件编写过程代码。

【例5-4】　修改例5-3中的窗体，使其增加一组字号选择单选按钮，并使用框架对窗体上的控件进行分组。

打开名为“frmExam_5_3”的窗体，先调整窗体中各个控件的布局，然后添加1个框架并在其内部增加2个单选按钮optSize(0)和optSize(1)，标题分别设为“20点”、“12点”。最后再将字体单选按钮组与字形选项组也放置在框架中。修改后的窗体布局如图5-6所示。

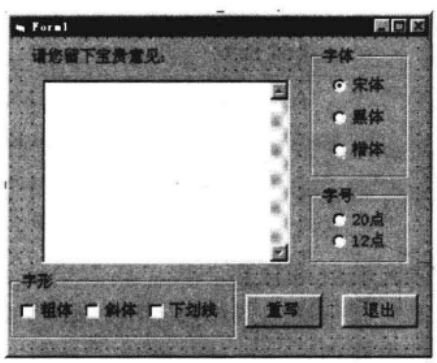


图5-6　窗体布局

思考：如何修改本例的事件过程代码以响应用户的选择？提示：使用 FontSize 属性可以改变文本框的字号。

5.3 列表框与组合框

单选按钮与复选框是供用户进行小范围选择的控件。如果选择的范围比较大，如学生选修的课程名称选择、国家名称的选择等就需要使用列表框或组合框控件了。列表框（ListBox）和组合框（ComboBox）在有限空间内提供了大量的选项，如果项目的数量超过了列表框或组合框所能容纳的数目，会自动出现滚动条。图 5-7 与图 5-8 分别表示了两种控件的应用示例。

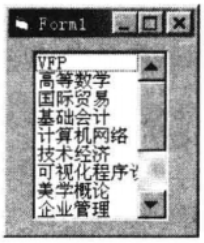


图 5-7 列表框



图 5-8 组合框的 3 种表现形式

组合框与列表框不同，根据 Style 属性的设置，它有 3 种不同的风格。

（1）下拉组合框

下拉组合框（Style 属性为 0）是默认的形式。程序运行时其下的列表框并不显示，只有在用户单击了其下拉箭头后才能显示。在选择这种形式时，既可以用键盘直接在文本框内输入，也可以打开列表框后进行选择（如 Word 中“格式”上的“字体”与“字号”下拉列表框就是典型的代表）。

（2）简单组合框

简单组合框（Style 属性为 1）在任何时候都显示其项目列表，因此没有下拉箭头。可以直接输入也可以在其下拉列表框中选择，这种格式一般使用不多。

（3）下拉列表框

下拉列表框（Style 属性为 2）形式上与下拉组合框相同，但与下拉组合框不同的是，它只能打开项目列表后选择列表中的项目，而不能直接在文本框中输入。这种格式一般用于项目是固定的，用户不能随意更改只能选择的场合（如 Word 的“页面设置”对话框中的“纸张大小”下拉列表框）。

从表现形式看，两种控件的区别并不大，组合框实际上可以看成是文本框与列表框的组合。从使用上看，列表框可以选择一个项目也可以选择多个项目，而组合框通常只能选择或输入一个项目，所以组合框更适合于空间不足而需要显示大量项目的时候。

5.3.1 列表框与组合框的设置

除了常规属性外，列表框和组合框还有一些特殊属性，如表 5-2 所示。

表5-2　列表框与组合框的常用属性及意义

属性名称	设置及意义	备　注
List	存放列表部分每一项内容的字符型数组,下标0~ListCount-1	可读写
ListIndex	返回程序运行时用户所选择的项目顺序号,其范围也在0~ListCount-1	运行时只读
ListCount	返回列表部分项目的个数	运行时只读
Selected	返回程序运行时列表框中某个项目是否被选中,True-选中,False-未选中	运行时只读
Sorted	指定列表部分项目是否按字母表的顺序排列,True-排序,False-不排序	默认为不排
Text	返回程序运行时用户所选择项目的文本内容,等于List(ListIndex)	运行时只读
MultiSelect	指明运行时用户能否进行复选:0-禁止,1-简单复选,2-扩展复选	只读
Style	指明显示风格:组合框如上所述,列表框0-标准,1-带复选框	只读

从表5-2可以看出,列表框与组合框的状态设置可分为3种情况。

(1) 属性为可读写

属于这种情况的只有List属性,它既可以在设计时预先将内容赋到项目中,也可以在运行时对指定的项目进行修改。如:

```
lstKc.List(1) = "经济学"        '将课程列表框lstKc的第2个项目内容改为"经济学"
```

(2) 属性为只读

属于这种情况的有3个:Sorted、MultiSelect和Style,它们在设计时一旦设置后,在程序中便不能修改只能引用。

(3) 属性为仅运行时只读

属于这种情况的较多,有4个:ListIndex、ListCount、Selected和Text。这类属性既不能在设计时设置,也不能在运行时设置,它们只能在运行时被引用。因为这些属性值在运行时时刻都在变化。

5.3.2　列表框与组合框的使用

1. 项目的增加

无论是列表框还是组合框,增加项目都有两种方法:一种是在设计时通过属性框中的List属性将有关项目输入进去,另一种是使用AddItem方法。前者主要是用于赋初值,后者使用了程序代码来实现对项目的动态增加,因此使用较多。

☞ 向属性框中添加多个项目时,换行不要直接按Enter键,而要按快捷键Ctrl+Enter。

为叙述方便,假设有一个列表框名称为"lstKc",其初值为"VFP"、"高等数学"、"基础会计"、"计算机网络"、"技术经济"和"可视化程序设计",如图5-9所示。



图5-9　列表框窗体

AddItem方法的格式是:

```
object.AddItem item [, index]
```

其中,object为列表框或组合框的名称,如lstKc;item是要加入到列表框中的项目内容,必须是字符型数据;index指明新增项目在列表中的位置,

第一个位置 index 为 0,可省略,默认放在列表中的最后。

例如,在图 5-9 的 lstKc 列表框中增加一门“美学概论”的课程,放在最后,可用以下代码:

```
lstKc. AddItem "美学概论"
```

2. 项目的删除

删除列表项目只能通过方法来实现。VB 提供了两种“方法”来删除项目。

(1) 删除选定项目

删除选定项目就是在程序运行时将用户所选择的项目从列表中清除,它使用 RemoveItem 方法,格式是:

```
object. RemoveItem index
```

该方法的语法格式与 AddItem 基本相似且更简单,它只给出需要删除的项目顺序号即 index 值就行了。

例如:lstKc. RemoveItem 6 命令可以将刚增加的“美学概论”删除。

如果要删除用户所选择的项目,可以使用以下命令:

```
lstKc. RemoveItem lstKc. ListIndex
```

ListIndex 是用户所选择的项目顺序号。

0 代表的是列表的第一个项目,而 ListCount - 1 则是指最后一个项目。

(2) 删除所有项目

如果要一次性地删除所有项目,可以使用 Clear 方法,其格式非常简单:

```
object. Clear
```

例如:执行 lstKc. Clear 命令将把图 5-9 的 lstKc 列表框中的所有项目全部删除。

3. 项目的访问

列表框与组合框都具有 Click 事件和 DblClick 事件,选择列表中的项目时将触发 Click 事件,而双击列表项目时将触发 DblClick 事件。为这两个事件编写一定的过程代码就可以访问列表中的项目了。当然,更常用的是将项目访问的代码放到其他控件的事件过程中(如某些按钮的 Click 事件中)。无论访问代码放在哪里,项目的访问需要引用表 5-2 中的某些属性才能实现。项目的访问一般有以下几种情况:

(1) 访问列表中的所有项目

可以用 List 属性来访问列表中的所有项目。因为 List 属性本身就是一个字符型的数组,列表中的每一个项目都是其元素。例如,可用下列格式来得到指定项目的内容:

```
lstKc. List( Index)
```

其中,Index 是待引用项目的顺序号。这种引用是一种绝对引用,与用户的选择无关。使用这种方法访问时要注意下标不要越界,即 Index 的值不能超过 ListCount - 1。

例如:lstKc. List(3)即是指列表框中的“计算机网络”课程。

(2) 访问用户选定的项目

程序中应用更多的是要访问在程序运行时由用户选定的项目,要实现这一点必须要判断选定项目所在的位置。用 ListIndex 属性即可判断选定位置并返回内容。例如,下面的命令可以返回 lstKc 列表框中选定项目的内容:

```
lstKc. List( lstKc. ListIndex)
```

置按钮的 Enabled 属性来实现。

② 两个列表框都要能实现复选操作，可以通过其 MultiSelect 属性设置，但对选中项目的访问不能放置在本身的 Click 事件中，而应放在相关按钮中。

③ 双击事件的实现只要在有关的事件代码中调用相应按钮的事件即可。

窗体上相关控件及其属性的设置如表 5-3 所示。

表 5-3 "课程选择"窗体的主要控件属性设置

控件名	属性名	取值	备注
列表框(lstKc)	MultiSelect Sorted List	True True 如图 5-9 中所示	备选课程列表框
列表框(lstBjkc)	MultiSelect	True	拟选课程列表框
命令按钮(cmdYc)	Enabled	False	初始禁止
命令按钮(cmdQy)	Enabled	False	初始禁止

有关的事件过程代码如下：

```
Option Explicit
Private i As Integer

Private Sub cmdExit_Click( )
    Unload Me
End Sub
```

```
Private Sub cmdQj_Click( )                    '"全部添加"按钮事件过程
    For i = 0 To lstKc. ListCount - 1
      lstBjkc. AddItem lstKc. List(i)         '将每个项目均添加到 lsrBjkc 框中
    Next i
    cmdTj. Enabled = False                    '设置按钮的状态
    cmdQj. Enabled = False
    cmdYc. Enabled = True
    cmdQy. Enabled = True
End Sub
```

```
Private Sub cmdQy_Click( )                    '移除全部事件过程
    lstBjkc. Clear                            '执行 Clear 方法
    cmdYc. Enabled = False                    '设置按钮状态
    cmdQy. Enabled = False
    cmdTj. Enabled = True
    cmdQj. Enabled = True
End Sub
```

```
Private Sub cmdTj_Click( )                    '添加事件过程
    For i = 0 To lstKc. ListCount - 1
```

```
        If lstKc. Selected(i) Then                    '如果第 i 项被选中,则将其添加到 lstBjkc 中
            lstBjkc. AddItem lstKc. List(i)
        End If
    Next i
    If lstBjkc. ListCount > 0 Then                    '只要 lstBjkc 中有项目,将按钮 Enabled 属性设为 True
        cmdYc. Enabled = True
        cmdQy. Enabled = True
    End If
End Sub
```

```
Private Sub cmdYc_Click( )                            '移除事件过程
    Dim bolSel As Boolean
    bolSel = True                                     '判断标志:有无选中的项目
    Do While bolSel And lstBjkc. ListCount > 0 '从列表框的最后一项检查起,直到第 1 项
For i = lstBjkc. ListCount - 1 To 0 Step - 1
            If lstBjkc. Selected(i) Then
                lstBjkc. RemoveItem lstBjkc. ListIndex          '移除所选择的项目
                bolSel = True
            Else
                bolSel = False
            End If
        Next i
      Loop
    If lstBjkc. ListCount > 0 Then                    '根据情况设置按钮状态
        cmdYc. Enabled = True
        cmdQy. Enabled = True
    Else
        cmdYc. Enabled = False
        cmdQy. Enabled = False
    End If
    cmdTj. Enabled = True
    cmdQj. Enabled = True
End Sub
```

```
Private Sub Command1_Click( )
    lstKc. RemoveItem lstKc. ListIndex
End Sub

Private Sub lstBjkc_DblClick( )                       'lstBjkc 的双击事件过程
    cmdYc. Value = True                               '直接调用移除过程代码
End Sub

Private Sub lstKc_DblClick( )                         'lstKc 双击事件过程
```

```
    cmdTj. Value = True                    '直接调用添加过程代码
End Sub
```

将本例命名为“frmExam_5_5”,调试运行通过后存放在 Vbxx 工程中。

① 本例中当选择多个项目时,双击事件还有效吗?

② 本例的代码并没有考虑在“拟选课程”中可能出现的课程重复问题。因此会出现一门课程被重复选择的问题,如何改正?

5.4 其他常用控件

除了前面已经讲述的 7 种控件以外,VB 所提供的常规控件还有以下几种:

① 水平滚动条(HScrollBar)与垂直滚动条(VScrollBar)。

② 定时器(Timer)。

③ 图片框(PictureBox)与图像(Image)。

④ 形状(Shape)与直线(Line)。

⑤ 数据(Data)。

⑥ 对象链接与嵌入(OLE)。

对于第③种与第⑥种控件将在本书的第 10 章中介绍,第⑤种控件将在第 9 章中重点介绍,而对于第④种的形状与线条控件在界面的设计中通常作为辅助的控件使用(如用于界面的美化、视觉区分等),因此其设计是非常简单的,读者完全可以自学。本节主要讲述前两种控件的使用。

5.4.1 滚动条

滚动条不仅能出现在文本类控件(文本框、列表框、组合框)的边缘以滚动文本内容。而且它还是一种相当不错的“模糊”输入与定位控件,当不需要精确设置数值,只要一个大概的范围时(如音箱的音量、处理某一事件所花的大致时间等),使用滚动条来进行输入是非常方便的。另外利用滚动条还可以实现微调按钮的功能(VB 中没有提供微调按钮控件)。

滚动条分为水平滚动条(HScrollBar)与垂直滚动条(VScrollBar)两种,除了走向不同外,其使用、设置均完全相同。每个滚动条都由 3 部分构成:2 个滚动按钮与 1 个滚动块。



由滚动条控件创建的滚动条与一般附加在文本控件上的滚动条是不一样的,这是因为附加在文本类控件上的滚动条是自动出现的。

1. 滚动条的属性设置

在程序中使用滚动条时当然也要对滚动条的某些属性进行必要的设置。滚动条的常用属性如表 5-4 所示。

表 5-4 滚动条的属性设置

属性名称	设置及意义	备注
Min	滚动块位于滚动条顶端或左端时,所代表的滚动条数值,-32 768 ~ 32 767 之间	可读写
Max	滚动块位于滚动条底端或右端时,所代表的滚动条数值,-32 768 ~ 32 767 之间	可读写

属性名称	设置及意义
LargeChange	在滚动条上空白处单击时,滚动条数值的增减量
SmallChange	单击滚动条两端的滚动箭头时,滚动条数值的增减量
Value	滚动条的当前值

滚动条在设计时除了一些常规的属性设置外,主要是针对前4个属性进行设置。Max Min的值根据实际的应用确定(如要设置某一种颜色值,可将Min设为0,Max设为255),一般况下Min应小于Max。

2. 滚动条的应用

包含滚动条的程序运行时,无论是单击滚动箭头或滚动条,还是用鼠标直接拖动滚动块都会引起滚动块位置的变化。因此,滚动条的应用关键是要随时监视滚动块的移动与滚动条的当前取值。

监视滚动块的移动通过Scroll事件和Change事件可以实现,当用鼠标直接拖动滚动块时将触发Scroll事件。当滚动条上的滚动块位置发生变化时(无论如何产生)将触发Change事件,本质上说只要滚动条的值发生变化就会引发Change事件。所以,通常都用Change事件来跟踪滚动条的状态变化。

获取滚动条的当前值非常简单,只要在适当的过程代码中取得滚动条的Value属性值即可。

【例5-6】 一般的计算机显示系统均支持24位的真彩色,24位的真彩色共有1 670多万种不同的颜色。在VB中通过RGB()函数可以产生这些不同的颜色,该函数的语法格式如下:

RGB(red, green, blue),其中red、green、blue分别表示红、绿、蓝三基色的值,其取值范围均为0~255。

本例要求使用3个滚动条(模拟三基色)和RGB()函数产生这些颜色并显示在一个矩形的背景上。

窗体的布局如图5-11所示,有关主要控件的属性及设置如表5-5所示(不含简单的控件)。

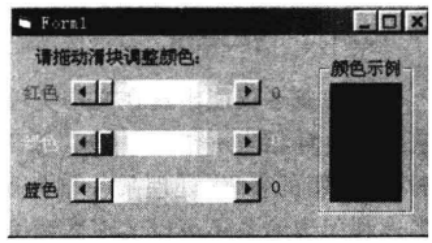


图5-11　例5-6运行结果

表5-5　例5-6窗体控件及属性设置

控件名称	属性名称	取值	备注
水平滚动条数组 hsbColor(0)~hsbColor(2)	Min	0	3个滚动条都采取同样的设置
	Max	255	
	LargeChange	32	
	SmallChange	8	
标签 lblColor(0)	ForeColor	红色	3个颜色标签的初始值均为0
标签 lblColor(1)	ForeColor	绿色	
标签 lblColor(2)	ForeColor	蓝色	
形状 shpReg	BackColor	黑色	三基色的值均取0

该窗体的代码非常简单,只要在滚动条的 Change 事件过程代码中输入以下指令即可实现本题的要求。

```
Private Sub hsbColor_Change(Index As Integer)
    lblColor(Index).Caption = hsbColor(Index).Value      '将每种颜色的值显示在相应的颜色标签上
    shpReg.BackColor = RGB(hsbColor(0).Value,_           '设置矩形框的背景色
        hsbColor(1).Value, hsbColor(2).Value)
End Sub
```

程序运行通过后以 frmExam_5_6 为名保存在 Vbxx 工程中。

5.4.2 定时器

定时器(Timer)是一种特殊的控件,它能有规律地以一定的时间间隔激发定时器事件(Timer)而执行相同的程序代码。定时器主要有两种功能:一是定时完成某项任务,二是进行某种后台处理。定时器在运行时并不可见,因而其在窗体上位置并不重要,通常只需在工具箱中双击即可完成创建。

定时器的设置比较简单,有两个属性。

(1) Interval(时间间隔)

Interval 属性反映了触发两次 Timer 事件的时间间隔,以毫秒(ms)为单位,其取值在 0 ~ 64 767 ms之间。取 0 表示不响应 Timer 事件,最大不能超过 64.8 s。该属性一般在设计时设置,当然也可以在代码中动态修改。



Interval 的值越小,则 Timer 事件响应越频繁,这样会降低系统的综合性能。因此除非必要,设置的时间间隔不能过小。

(2) Enabled

该属性与一般控件的 Enabled 属性含义相同,取 True 表示定时器开始工作(默认状态),取 False 表示定时器停止工作。Enabled 属性通常在程序中动态设置。

定时器的状态设置完成后,一般只要在其 Timer 事件中加入需要完成的功能代码,如果需要的话,在相关的按钮事件过程中再加上启动或停止定时器代码就可以了。

【例 5-7】 修改例 5-6 的窗体,使其能每隔 0.5 s 随机显示一种颜色,并加上“启/停”按钮。

打开 frmExam_5_6 的窗体,在窗体上增加 1 个定时器控件,再增加 3 个命令按钮,分别为“开始”、“停止”和“退出”。

定时器的属性作如下设置。名称:tmrColor,Enabled:False,Interval:500。按钮的有关属性设置及代码比较简单,在此不再赘述。

在定时器的 Timer 事件过程中输入以下代码:

```
Private Sub tmrColor_Timer()
    Dim i As Integer
    For i = 0 To 2
    '以当前时间为随机种子,产生 0 ~ 255 之间的随机数作为某个颜色的值。
    hsbColor(i).Value = Int(255 * Rnd(Time))
```

```
        Next i
        shpReg. BackColor = RGB( hsbColor(0) Value, _
        hsbColor(1). Value, hsbColor(2). Value)
    End Sub
```

程序运行后，单击“开始”按钮，每隔0.5 s在矩形区域内就会显示一种不同的颜色（图5-12），单击“停止”按钮，定时器即停止计时。将程序以frmExam_5_7为名存放在Vbxx工程中。

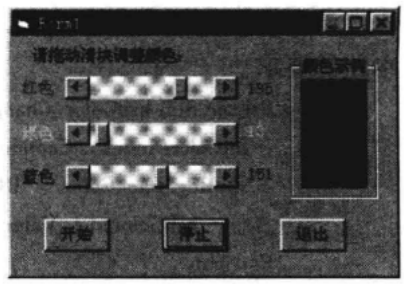


图5-12　随机颜色的产生

5.5　窗体控制与设计

通过第2章对窗体常用属性与事件的介绍可知，窗体（Form）是设计VB应用程序的一个基本平台，所有的VB控件都要放置在窗体之上，用户正是通过操作窗体上的各种控件来使用应用程序的。任何应用程序至少有一个窗体，从表现形态来看，大部分窗体在运行时与Windows应用软件中的“对话框”相似。

5.5.1　窗体的运行与控制

窗体作为一种特殊的对象，有它自己的事件和方法，通过为窗体的事件编写代码或调用窗体的方法既可以控制窗体的运行，又能完成用其他对象难以实现的工作。尽管在一般情况下，用户针对窗体的操作并不多，但由于窗体是控件的载体，是程序运行的场所，这样有些程序运行时所必需的初始化操作或程序退出后的“善后处理”，还是需要通过窗体的相关事件或方法来实现。

1. 多重窗体

对于一般的应用程序而言，总要包含多个窗体以完成不同的任务。在多重窗体中，每个窗体都有自己的界面与程序代码，都是以独立的.frm文件保存在工程中。

利用“工程”菜单上的“添加窗体”命令或单击工具栏上的“添加窗体”按钮都可以创建一个新的窗体。

当多重窗体设计调试完成后，必须将这些窗体有机地联系起来，以形成完整的应用程序。一般来说，如果某个窗体是独立的，它只附属于应用程序，那么就可以在应用程序相应的菜单项或工具栏中使用调用命令来调用该窗体；如果某个窗体附属于另外一个窗体，那么调用该窗体的命令就应该放在另外窗体的相应事件代码中。当一个窗体使用完成后应该将其释放。调用一个窗体可以使用下述的Show方法，而释放一个窗体则可以使用Unload命令。

2. 设置启动窗体

既然一个应用程序通常都由若干个窗体构成，那么控制窗体运行的首要任务就是要决定当程序一开始运行时先执行哪个窗体，这个最先执行的窗体就称为“启动窗体”。关于启动窗体的设置方法在前面已经介绍，这里要说明的是，程序的启动也可以不通过窗体实现，而通过调用一个特殊的过程来启动程序的运行。但这个过程的名称必须是Main()，而且它必须位于标准模块中。例如可以创建一个这样的启动过程：

```
Sub Main( )
    frmPass. Show   '显示密码验证窗体
End Sub
```

要使该过程先运行,与设置启动窗体一样,选择〖工程〗→〖工程属性〗菜单命令,在“通用”选项卡的“启动对象”中将“Sub Main”选为启动对象即可。

在多重窗体的设计中,要调试指定的窗体必须将 VB 的启动窗体改为指定的窗体。

3. 窗体的生命周期

窗体的生命周期是指窗体从开始运行到运行结束所经历的不同状态。在整个生命周期中,窗体具有 4 种状态:创建状态、加载状态、显示状态和卸载状态。这 4 种状态是通过触发 4 种不同的事件实现的。

(1) 创建状态

使用 Load 命令加载窗体时,窗体首先处于创建状态,将触发 Initialize 事件,所以 Initialize 事件是窗体处于创建状态的标志。处于创建状态的窗体仅作为对象已经存在,但其上的控件还没有生成。利用这种状态可以将程序需要最先执行的代码放置在 Initialize 事件中,但要注意的是在该事件中不能引用窗体上的任何控件。Load 命令的语法为:

```
Load frmName                'frmName 为窗体的名称
```

(2) 加载状态

Load 事件是窗体处于这种状态的开始标志,窗体一旦进入加载状态将执行 Load 事件的过程代码,此时窗体上所有的控件均被创建并加载,窗体也已经形成,只不过是处于不可见状态。

由于 VB 程序在执行时会自动装载启动窗体,因此没有必要使用显式的加载语句。但要注意的是,当系统自动装载窗体时也会引发窗体的 Load 事件。

(3) 显示状态

使用窗体的 Show 方法可以使窗体进入显示状态。在显示时,系统将首先检查窗体是否已经处于加载状态,如果没有加载,则先加载然后显示。一旦窗体显示,用户就可以与窗体进行交互了。

尽管一个窗体需要经过创建与加载两种状态后才能进入显示状态,但在 VB 的应用程序中要使一个窗体变成显示状态只要调用该窗体的 Show 方法即可,因为调用 Show 方法将首先创建窗体和加载窗体,然后才是显示窗体。Show 方法的格式如下:

```
object. Show [style]
```

其中,object 显然是窗体的名称;style 是可选项,决定了窗体的模式,该项取 0(默认值)表示窗体是无模式的窗体(不限制将焦点转移到其他窗体上),如该项取 1 表示窗体是模式窗体(在本窗体卸载或隐藏之前不能将焦点转移到其他窗体上)。

例如,一个应用程序在密码验证窗体自动加载后,如果用户的密码输入正确,将需要调用程序的主窗体(frmMain)。只要在密码验证窗体的“确定”按钮的 Click 事件过程中加上如下的命令即可:

```
frmMain. Show
```

☞ 将需调用窗体的 Visible 属性,设为 True 也可以显示窗体。

如果一个窗体暂时不再需要,可以考虑采用两种措施:

① 将窗体卸载。使用 Unload 命令可以将一个窗体卸载,如在前述的大部分例子中的“关闭”按钮中加入的“Unload Me”语句即表示将当前窗体卸载。显然将“Me”换成某一个处于加载

状态的窗体也可将其卸载。

② 将窗体隐藏。隐藏窗体的命令是调用窗体的 Hide 方法，其使用与 Show 方法相同。当一个窗体被隐藏后，该窗体将变成不可见状态，且窗体的 Visible 属性被置为 False。要注意的是，处于隐藏状态的窗体并未卸载，如果再调用 Show 方法，则该窗体又会显示。



窗体从隐藏状态变为显示状态并不触发 Load 事件，Load 事件在整个窗体的生命周期内只运行一次。

显示状态下的窗体有两个十分重要的事件，即 Activate 事件和 Deactivate 事件。当一个窗体变成活动窗体（窗体或窗体中的某个控件具有焦点）时将触发 Activate 事件，而当另一个窗体或应用程序激活时将触发 Deactivate 事件。Activate 事件通常用于窗体的初始化，如将焦点首先设置在某一个文本框中；Deactivate 事件通常用于窗体的结束工作，如将所修改的数据保存到文件或数据库中。

（4）卸载状态

窗体的卸载是指从内存中清除窗体，回收窗体所占用的资源。如果窗体已经不再使用时应及时将窗体卸载。要将窗体卸载只要使用前述的 Unload 命令即可。执行该命令与窗体是否可见无关，也即无论窗体是否可见都可用该命令将指定的窗体卸载。

在程序中卸载窗体时一定要特别小心，因为一旦窗体被卸载，与该窗体有关的所有数据将全部清除，所以一定要将需要保存的数据保存完毕后才能卸载窗体。但有时候，窗体的卸载是由于用户的误操作而引起的（如用户单击了“关闭”按钮），如何才能捕捉到这种误操作呢？VB 安排了一个特殊而又重要的事件 QueryUnload 事件，该事件在窗体卸载前会触发，这样为停止卸载窗体提供了机会。QueryUnload 事件的语法是：

```
Private Sub Form_QueryUnload(cancel As Integer, unloadmode As Integer)
```

在该事件中包含两个参数。

① cancel：确定是否卸载窗体，如为 0 则卸载，不是 0 则停止卸载。

② unloadmode：窗体将卸载的原因，它有 0～5 六个值，其含义分别如下。

- 0 – 双击窗体图标或选择控制菜单中的“关闭”命令。
- 1 – 在代码中使用 Unload 语句。
- 2 – 当前 Windows 操作结束。
- 3 – 利用 Windows 任务管理器关闭应用程序。
- 4 – MDI 窗体正在关闭。
- 5 – 窗体的所有者正在关闭。

该事件最简单的用法是无论使用何种方法来关闭窗体，均提示用户，以防误操作。

例如，在某个窗体的 QueryUnload 事件代码中输入如下代码：

```
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
    If MsgBox("确实要关闭本窗体吗?", vbQuestion + vbYesNo, Me.Caption) = vbNo Then
      Cancel = True              '禁止关闭
    End If
End Sub
```

程序运行后，只要关闭窗体就会弹出“确实要关闭本窗体吗?”的消息框，如选择 No，则不关闭，如选择 Yes，则窗体正常关闭。

如果使用了 END 语句或在 VB 选择〖运行〗→〖结束〗菜单命令来结束应用程序，窗体是不会触发 QueryUnload 事件的。

使用窗体的 Unload 事件也可以实现以上目的。QueryUnload 事件是在任一窗体卸载之前在所有窗体中发生的，而 Unload 事件是在每个窗体卸载时发生的。

Show 方法、Hide 方法和 Unload 命令在多重窗体的设计中是非常重要的，在 VB 的应用程序设计中正是通过这些方法或命令才得以将多个窗体连成一个完整的系统的。

【例 5-8】 多重窗体设计示例：设计一个具有密码验证、增加用户、修改密码 3 个窗体的多重窗体程序。

(1) 密码验证窗体

在 VB 的“添加窗体”对话框中有一个“登录对话框”，选择它以后会自动在当前工程中添加一个简单的具有密码验证功能的登录对话框。在此基础上进行必要的修改即可满足本题的要求。

添加后命名窗体为“frmLogin_5_8”，首先在其上增加 2 个按钮，即“修改密码”按钮和“增加用户”按钮，增加后如图 5-13 所示。

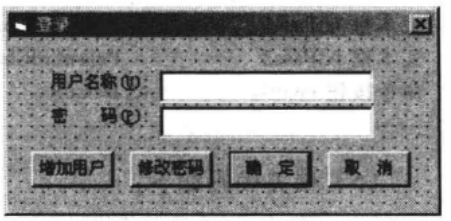


图 5-13 登录验证窗体

为了能实现密码修改和增加用户的功能，需要对代码进行必要的修改：

① 设置两个公用变量数组 strName(1 To 4)和 strPass(1 To 4)以存放不同的用户名称及其对应的密码(最多只能存放 4 位用户)。这两个数组必须放置在标准模块中。另外，在标准模块中再设置两个公用的整型变量 intUser、intNum 用来存放当前用户名和用户序号。

② 当用户单击“修改密码”按钮时，要判断用户及其密码的合法性。如果是合法的，则调用“修改密码”窗体(frmExam_5_81)。

③ 当用户单击“增加用户”按钮时，要求用户必须是管理员用户，而且当前用户的数目不能大于 4。假设管理员用户的用户名与初始密码均是“SYSTEM”。

(2) 密码修改窗体(frmExam_5_81)

密码修改窗体的布局如图 5-14 所示。该窗体比较简单，只有 2 个文本框与 2 个按钮。要求密码输入两次，相同后才能写入公用变量中。写入代码放置在“确定”按钮中。

(3) 增加用户窗体(frmExam_5_82)

增加用户的窗体布局如图 5-15 所示。它所要完成的是在增加用户时，用户名不能为空，两次输入的密码必须一致。否则给出适当的提示信息要求重输。

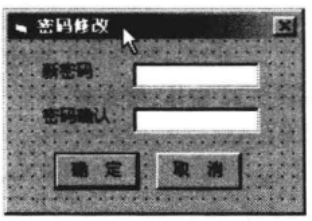


图 5-14 “密码修改”窗体

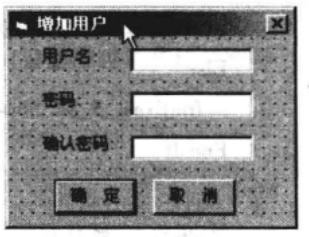


图 5-15 “增加用户”窗体

（4）为使用户不能对窗体进行最大化或最小化，将3个窗体的MaxButton、MinButton属性均设为False。窗体与控件的主要属性设置如表5-6所示。

表5-6　例5-8相关属性设置

对象名称	属性名称	设置值	备注
frmExam_5_8	Caption	登录	登录验证窗体
文本框 txtPassWord	PasswordChar	*	密码显示字符
命令按钮 cmdOk	Default	True	默认按钮
frmExam_5_81	Caption	密码修改	密码修改窗体
文本框 txtNewpass1	PasswordChar	*	
文本框 txtNewpass2	PasswordChar	*	
命令按钮 cmdOk	Default	True	
frmExam_5_82	Caption	增加用户	增加用户窗体
文本框 txtPass1	PasswordChar	*	
文本框 txtPass2	PasswordChar	*	
命令按钮 cmdOk	Default	True	

（5）相关代码如下：

```
'frmExam_5_8 窗体代码
Option Explicit
Public LoginSucceeded As Boolean                    '登录成功标志
-------------------------------------------------------------------------
Private Sub cmdAdd_Click()                          '增加用户事件过程
    Dim i As Integer
    '判断用户是否为管理员
    If txtUserName.Text = "SYSTEM" And txtPassword.Text = "SYSTEM" Then
        For i = 1 To 4                              '判断用户数
          If strName(i) = "" Then
            intNum = i
            Exit For
          End If
        Next i
        If intNum = 4 And strName(4) <> "" Then
          MsgBox ("用户已满,不能增加",,"登录")
        Else
          frmExam_5_82.Show 1                       '调用 frmExam_5_82 窗体,模式显示
        End If
    Else
        MsgBox ("只有管理员才能增加用户,请返回!",,"登录")
    End If
```

```
End Sub

Private Sub cmdCancel_Click( )                    '取消按钮事件过程
    LoginSucceeded = False                        '登录成功标志:不成功
    Me. Hide                                      '隐藏本窗体
End Sub

Private Sub cmdChange_Click( )                    '修改密码窗体事件过程
    cmdOK. Value = True                           '显式调用确定按钮事件过程以确定用户的合法性
    If LoginSucceeded Then                        '如是合法用户则调用 frmExam_5_81 以供密码修改
        frmExam_5_81. Show
        Me. Hide
    End If
End Sub

Private Sub cmdOK_Click( )                        '确定按钮事件过程
  '检查正确的用户名与密码
  Dim i As Integer
  For i = 1 To 4
  If txtPassword. Text = strPass(i) And  _
    txtUserName. Text = strName(i) _
      And txtUserName. Text < > "" Then
      LoginSucceeded = True
      MsgBox ("密码正确,欢迎进入!", , "登录")
      intuser = i
      Me. Hide
      Exit For
    Else
      LoginSucceeded = False
    End If
    Next i
    If Not LoginSucceeded Then
      MsgBox ("无效的密码或用户名,请重试!", , "登录")
      txtPassword. SetFocus                       '密码或用户名不正确,重新输入
      SendKeys "{Home} + {End}"                   '模拟键盘输入
    End If
End Sub

Private Sub Form_Load( )                          '窗体加载时设置管理员名称与密码
    strName(1) = "SYSTEM"
    strPass(1) = "SYSTEM"
End Sub

'修改密码 frmExam_5_81 窗体代码
Private Sub cmdCancel_Click( )                    '取消事件过程
```

```
    Unload Me                                    '卸载本窗体
    frmExam_5_8. Show                            '重新显示登录窗体
End Sub
```

```
Private Sub cmdOK_Click( )                       '确定按钮事件过程
If txtNewPass1. Text = txtNewPass2 And txtNewPass2. Text < > "" Then
                                                 '比较两次输入密码是否一致
    strPass( intuser) = txtNewPass1
    Unload Me
    frmExam_5_8. Show
  Else
    MsgBox ("两次输入的密码不同,请重新输入", , "密码修改")
    txtNewPass1. SetFocus                        '两次输入不一致,重新输入
    SendKeys "{home} + {end}"
  End If
End Sub
```

```
'增加用户 frmExam_5_82 窗体代码
Private Sub cmdCancel_Click( )
    Unload Me
End Sub
```

```
Private Sub cmdOK_Click( )
    If txtNewuser. Text < > "" And txtPass1. Text = txtPass2. Text Then
      strName( intNum) = txtNewuser. Text    '将增加的用户与密码写入公用变量
      strPass( intNum) = txtPass1. Text
    Else
      MsgBox "用户名缺或密码错! 请重新输入", , "增加用户"
    End If
    Unload Me
    frmExam_5_8. Show
End Sub
```

5.5.2　窗体的设计原则

用户在使用应用程序时,绝大部分的时间在与各种各样的窗体打交道。而开发应用程序中首先要做的也是设计窗体的布局与界面,因此在窗体设计时应该遵循一定的原则以便设计出美观、易用的应用程序界面。

简单地说,窗体在设计时应注意以下原则。

(1) 遵循 Windows 应用软件界面的一般原则

Windows 之所以形成今天独霸天下的局面,很大程度上得益于其应用软件友好而又统一的使用界面。当学会了某种应用软件后,很容易学会使用其他应用软件。如果开发的应用程序的界面与这些软件的界面相差甚远,人们使用时显然会无所适从。因此在开发时应该多观察一些

通用的软件，从中不难找到一些共性的东西，如工具栏、状态、工具提示和各种对话框等。

（2）界面元素的一致性

界面元素的一致性是指在构成应用程序的多个窗体中所使用的控件类型、尺寸要相对统一，界面上控件分组要一致，字体、颜色等属性也应尽量前后统一。

（3）控件位置的层次性

尽管各个窗体所采用的控件个数及类型是不确定的，但控件在窗体上的摆放要注意有一定的层次。也就是说，控件的位置应符合人们一般的阅读习惯。比如，重要的元素应放置在窗体的左上部位，按钮一般应放在窗体的右下部位，对窗体元素的操作也应该尽量遵循从上到下、从左到右的原则。

将界面上的元素按功能或逻辑关系进行必要的分组也是非常重要的，这样将有助于用户对应用程序的操作。

对于控件数目较多的窗体，在设计时要注意使用适当的空白空间以突出元素和改善可用性。如果控件在窗体上“济济一堂”，必然会导致界面的杂乱无章，这样操作起来也会变得非常困难。

（4）保持界面的简单明了

界面设计中最重要的原则应该是简单化。所谓“简单化”就是指用最少的控件、最少的操作步骤能够实现指定的功能。

① 不要总是想用界面来模仿真实的对象。比如，在设计“学生基本情况表”的输入界面时，就不能将界面设计成一张真实的“学生基本情况表”，而应该对其进行深入的分析，从而采用简单明了的界面来实现信息的输入。

② 避免滥用颜色与图像。使用一定的颜色或图像可以增加视觉上的感染力，但过多而又杂乱的颜色或图像，一方面会直接影响用户的情绪，还容易造成“喧宾夺主”。因此，在设计中还是应该采用一些柔和的、中性化的设计。

③ 图片与图标使用要统一。图片与图标的选择应该尽量与传统的 Windows 软件相一致。

④ 字体选择要一致。字体是用户界面的重要组成部分，在选择字体时应选用那些在不同分辨率、不同类型的显示器上都能容易阅读的字体。要尽量使用标准字体，而且字体的种类不能太多，一般而言不宜超过两种。

5.6 案例三的设计

通过本章的学习，相信读者已经基本掌握了 VB 一些典型控制的使用方法。使用这些控件基本可以构造满足一般需要的对话框或窗口。本节通过案例三的制作过程将使读者进一步理解并能综合运用这些 VB 控件。

案例三的要求如下：

① 可以查询不同的项目，如姓名、学号、成绩和课程等。

② 可以选择查询的条件，如小于 60 分、课程为“计算机网络”等。

③ 能生成复杂的查询条件，如“计算机 00 班”的“高等数学”课程不及格的学生。

④ 具有简单的容错处理，如数据的合法性检查、控件的动态控制等。

设计步骤如下。

(1) 确定窗体风格及控件类型

由于是一个查询生成器类型的窗体,因此将窗体设计成固定大小(BorderStyle 取 1),无最大化和最小化按钮,并选择一种图标作为窗体的图标。

控件的选择主要考虑以下情况。

① 查询项目使用组合框 cboCxxm,初始值分别是:学号、姓名、班级、课程、学年学期、成绩。

② 关系运算符也使用组合框 cboYsh,为便于使用,其列表项目采用了中文化的运算符,分别为:等于、小于、小于等于、大于、大于等于、不等于。由于程序代码并不支持这种写法,在程序代码中设置一个数组 strYsh(5),其项目与之相对应为"=","<","<=",">",">="和"<>"。

③ 输入具体值采用一般的文本框 txtCxxmz。对成绩这个查询项目另外还设置了滚动条 vsbCjxq 以便于微调输入在文本框中的数值,这个滚动条在一般情况是隐藏的,只有用户选择了该查询项目后才显示。

④ 为输入复杂条件使用了 2 个单选钮 optLjysh("与"和"或"),考虑到需要将条件取反,设置了一个检查框 chkBds。

⑤ 当用户的条件输入完成后,使用"添加"按钮可将输入的条件变成一个合法的 VB 逻辑表达式,并显示在表达式文本框中(txtCxbds),该文本框为只读(Locked 属性为 True),不能编辑。

(2) 窗体布局设计

根据以上设想,在 Vbxx 工程中新建一个窗体(frmExam_5_9),将有关的控件绘制在窗体上,如图 5-16所示。本例控件的属性设置并不困难,请读者根据图 5-16 对有关属性进行必要的设置。

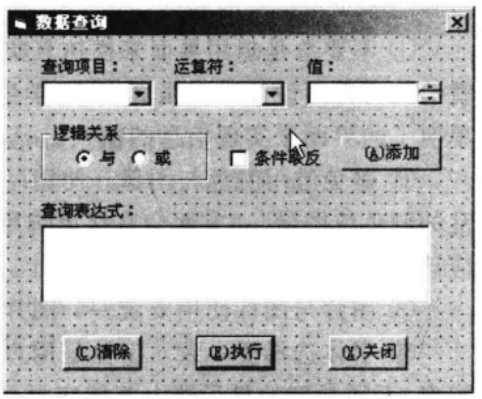


图 5-16　"数据查询"窗体布局

(3) 代码设计

本例题的代码设计主要集中在以下几点:

①"添加"按钮的 Click 事件过程。该事件过程的主要任务是将查询项目框、运算符框和值框中的值提取出来并经过适当的变换,形成一个有意义的关系表达式,这是其一。第二,要根据查询表达式框中已有表达式的情况与逻辑关系和检查框的值,进行合法的组合以形成合法的逻辑表达式。第三,要在形成表达式之前对输入值的情况进行判断,如果没有输入,会弹出提示的消息框。

② 窗体的 Load 事件。该事件主要完成在窗体装载时首先要进行的初始化工作,如运算符数组的初始化。当然对查询项目框、运算符框中的项目也可在此用 AddItem 方法进行赋值。另外还要设置滚动条的有关属性(主要是 3 个属性:Max、Min 和 SmallChange)。

③ 控制滚动条的动态显示。由于只有在查询"成绩"时才使用滚动条,因此可以在查询项目组合框的 Click 事件中进行判断。当选择了这项之后,设置滚动条的 vsbCjxq 的 Visible 属性为 True,否则设置为 False。

④ 滚动条的 Change 事件。当用户单击滚动条时将会发生该事件,通过该事件代码准确地

改变输入值文本框中的相应值。

⑤ 其他。除了以上代码以外,还要编制命令按钮的 Click 事件、输入值框的 Change 事件(用于设置滚动条的 Value 值)和输入值框的 Validate 事件(检查数据输入的合法性)等。

主要代码如下所示:

```
Option Explicit
Dim strYsh(5) As String                          '运算符数组
Dim strBds, strType, strLjysh As String
Dim i As Integer
```

```
Private Sub cboCxxm_Click()                      '查询项目框的单击事件
  If cboCxxm.Text = "成绩" Then
      vsbCjxq.Visible = True                     '显示滚动条
  Else
      vsbCjxq.Visible = False                    '隐藏滚动条
  End If
  txtCxxmz.Text = ""                             '清除输入框中的数据
End Sub
```

```
Private Sub cmdAdd_Click()                       '"添加"按钮事件过程
    If txtCxxmz = "" Then                        '判断有无输入数据
      MsgBox("请输入适当的值", , "数据查询")
      txtCxxmz.SetFocus                          '置焦点
    Else
      If cboCxxm.ListIndex <= 4 Then             '根据所查询的项目设置不同的定界符
        strType = Chr(34)                        '除成绩外,定界符都设为双引号
      Else
        strType = ""                             '如是成绩则不设定界符
      End If
      If optLjysh(0).Value Then                  '判断逻辑关系,转换成逻辑运算符
        strLjysh = " And "
      Else
        strLjysh = " Or "
      End If
      '取出各框中的数值并进行转换形成表达式字符串
      strBds = cboCxxm.List(cboCxxm.ListIndex) + _
       " " & strYsh(cboYsh.ListIndex) _
       & " " & strType & txtCxxmz & strType
      If chkBds.Value Then                       '检查有无取反,如取反则加上必要的字符
          strBds = " Not " + "(" + strBds + ")"
          chkBds.Value = False
      End If
      If txtCxbds.Text = "" Then                 '判断表达式框中的数据情况
```

```
                txtCxbds. Text = strBds                    '为空,直接送出
            Else
                txtCxbds. Text = txtCxbds. Text + strLjysh + strBds        '不为空,则进行连接
            End If
        End If
    End Sub
```

```
    Private Sub cmdGroup_Click(Index As Integer)           '命令按钮组单击事件
        Dim intErr
        Select Case Index
        Case 0
            txtCxxbds. Text = ""                           '清除表达式框
        Case 1
        '此处填入执行代码
        Case 2
            Unload Me
        End Select
    End Sub
```

```
    Private Sub Form_Load()                                '窗体的装载事件
        strYsh(0) = " ="
        strYsh(1) = " <"
        strYsh(2) = " <="
        strYsh(3) = " >"
        strYsh(4) = " >="
        strYsh(5) = " <>"
        With vsbCjxq                                       '对成绩项的滚动条变化范围进行设置
            .Max = 0
            .Min = 100
            .SmallChange = 1
      End With
    End Sub
```

```
    Private Sub txtCxxmz_Change()                          '输入值文本框的 Change 事件
        Dim sincj As Single
        '如果查询项为"成绩",则根据用户输入的成绩设置滚动条的当前值
        If vsbCjxq. Visible And cboCxxm. Text = "成绩" Then
            sincj = Val(txtCxxmz. Text)
            vsbCjxq. Value = sincj
        End If
    End Sub
```

```
    Private Sub txtCxxmz_Validate(Cancel As Boolean)       '输入值文本框的 Validate 事件
        '根据不同的查询项目设置不同的数据合法性检查项,此处只检查了学年学期项
```

```
    If cboCxxm. Text = "学年学期" And _
       Len(txtCxxmz. Text) < > 6 And Mid(txtCxxmz. Text, 5, 1) < > "/" Then
        Label5. Caption = "格式为 2000/2"
        Cancel = True                                    '保持焦点
    End If
End Sub
```

```
Private Sub vsbCjxq_Change( )                            '滚动条的 Change 事件
    txtCxxmz. Text = Trim(Str(vsbCjxq. Value))
End Sub
```

(4) 程序调试并运行

将以上代码输入,进行适当的调试,完成后即可运行。例如,要查询"计算机 00"班"高等数学"课程成绩不低于 60 的情况,可进行本章图 5-1 所示的选择。

习 题 5

1. 试总结控件中有哪些常用的属性与事件? 其含义与用法如何?
2. 控件默认的 Tab 键顺序是什么? 如何改变?
3. 什么情况下会产生 Click 事件?
4. 试说明在以下动作中会引发哪些事件?

(1) 在文本框中输入字符。

(2) 文本框中的内容发生变化。

(3) 某一个控件失去焦点时。

5. 如何形成控件数组? 编写控件数组的事件过程与编写单个控件的事件过程有何不同?
6. 组合框有几种形式? 有何异同? 通过什么属性值来改变?
7. 能使用文本框与列表框来实现组合框吗?
8. 在引用一个列表框(List1)中当前选定的项目时,使用了如下语句:

```
List1. List(ListIndex)
```

试判断该语句是否正确? 正确的引用方式是什么?

9. 列表框与组合框有哪些常用方法? 其功能是什么?
10. 完善例 5-9,使在"拟选课程"中不出现重复的课程。
11. 如何设置窗体的外观状态?
12. 何谓启动窗体? 如何设置?
13. 在窗体的生命周期中有几种状态? 分别涉及哪些事件?
14. 窗体设计一般应遵守什么原则?
15. 设计一个"学生基本情况调查表"的窗体如图 5-17 所示,当用户单击了"提交"按钮后,输入的信息将显示在下方的文本框中,单击"清除"按钮将清除文本框中的内容。
16. 设计一个倒计时程序,运行效果如图 5-18 所示。运行时单击"设置"按钮,弹出 InputBox 输入框供用户输入倒计时的时间(以分钟为单位)。单击"开始"按钮则在文本框内显

示剩余的时间数(以时:分:秒的格式),同时滚动条上的滚动块同步随时间自左向右移动。

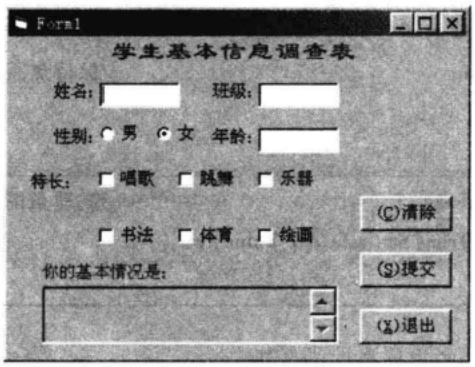


图5-17



图5-18

17. 设计一个多重窗体程序,要求如下:

(1)“成绩数据处理”窗体上有3个按钮:“成绩输入”、“成绩计算”、“结束”。

(2)单击“成绩输入”按钮时弹出“成绩输入”窗体,该窗体上放置几个文本框和标签用于输入各门课程的成绩,课程名称自定。其上还有一个“返回”按钮,用于返回到“成绩数据处理”窗体。

(3)单击“成绩计算”按钮时弹出“成绩计算”窗体,在该窗体上显示学生的平均成绩与总成绩,同样设置一个“返回”按钮。

第6章　键盘与鼠标的控制

学习目标

掌握在程序中处理键盘、鼠标的各种输入；熟悉与键盘、鼠标相关的事件；能够编写简单的拖放程序。

在计算机的操作系统中有许多系统资源，如键盘、鼠标、打印机和屏幕等。在程序中如能合理地使用这些系统资源，就可以最大限度地发挥系统的灵活性，对编写专业的应用程序会有很大的帮助。本章主要介绍系统资源中最为重要的键盘与鼠标的控制与编程。

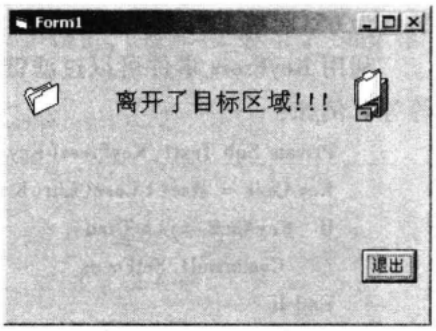


图6-1　文件复制动画演示

【案例四】 一个有趣的拖放动画。

在 Windows 系统中都支持各种拖放操作，最为典型的是对各种资源图标的直接拖放与排列。本案例演示了文件的复制功能，如图6-1所示。在文件夹的图标上按下鼠标，就会出现一个文件图标，将其拖放到右边的文件柜上，此时文件柜上的抽屉打开，松开鼠标后文件消失，同时抽屉关上，下方显示“文件已经接收”的提示。在拖动过程中，中间的标签文字内容会随着鼠标位置的不同显示不同的信息。

6.1　键盘的控制

键盘一直是计算机最主要的输入设备，VB 提供了3种键盘事件用于对键盘的控制（表6-1）。

表6-1　键盘事件

键盘事件	触发原因
KeyPress	按下和松开一个 ANSI 键时发生
KeyDown	按下任意键时发生
KeyUp	松开任意键时发生

只有获得焦点的对象才能接受键盘事件。对于键盘事件，只有当窗体为活动窗体且其上所有控件均未获得焦点时，窗体才获得焦点。

键盘的3个事件并不互相排斥。当按下一个键时将触发 KeyPress 和 KeyDown 事件，而松开此键后将触发 KeyUp 事件。

6.1.1　KeyPress 事件

按下和松开一个 ANSI 键时将触发 KeyPress 事件。该事件可用于窗体、复选框、组合框、列表框、命令按钮、图片框和文本框等大多数控件。当一个控件或窗体具有焦点时，该控件或窗体将接受从键盘上输入的信息。例如，当一个文本框具有焦点时，从键盘上输入的任何信息都将显示在文本框中。

KeyPress 事件过程的一般格式为：

```
Private Sub 对象名_KeyPress(KeyAscii As Integer)
 …
End Sub
```

该事件过程有一个参数 KeyAscii，它可以识别按键的 ASCII 码，例如，按下“A”键，KeyAscii 的值为 65，而按下“a”时，其值则为 97，等等。

利用 KeyPress 事件可以过滤键盘的输入、识别用户是否按下特定的键、实现大小写的转换等等。例如：

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
KeyAscii = Asc(UCase(Chr(KeyAscii)))
If KeyAscii = 13 Then
    Command1.SetFocus
End If
End Sub
```

该程序一方面将用户输入转换成大写字母，另外还用于检测用户在文本框中输入时是否按下了 Enter 键，如按下 Enter 键（ASCII 码值为 13）则将焦点移到命令按钮上。

☞ 也可以用 VB 所提供的“键代码常数”来代替相应的 ASCII 码值（如 Enter 键为 vbKeyReturn。各种键代码常数可通过在“对象浏览器”中搜索“Key Code Constants”得到各个键所对应的常数。

☞ 可以在程序中直接修改 KeyAscii 的值，这样在控件上显示的就不是用户实际的输入，而是修改后的值。特别是如果将 KeyAscii 值设为 0，则会封锁键盘，使对象接受不到字符。

6.1.2　KeyDown 与 KeyUp 事件

并不是所有的按键 KeyPress 事件都能检测，对于不能检测的按键可以通过 KeyDown 和 KeyUp 事件来控制。

与 KeyPress 事件不同，KeyDown 和 KeyUp 事件返回的是键盘的直接状态，换言之，KeyDown 和 KeyUp 事件所返回的是所按键的键代码（KeyCode）。

KeyDown 和 KeyUp 事件过程的格式如下：

```
Private Sub 对象名_KeyDown|KeyUp(KeyCode As Integer, Shift As Integer)
…
End Sub
```

事件带有两个参数：

(1) KeyCode

它是按键的实际键代码,键盘上所有的键位都有。该码是以“键”为准,也就是说,一个键只有一个 KeyCode,不分大写和小写,也不分上档下档键。对于字母键,则一律返回该键所对应的大写字母的 ASCII 码作为 KeyCode;对于具有上下档的键,则返回下档字符的 ASCII 码;而对于数字键来说,打字键区的数字键和数字键区的数字键返回不同的 KeyCode。对于不同键位的 KeyCode,可以通过该事件方便的求出(见后面的实例)

(2) Shift

这是一个反映 3 个控制键(Shift、Ctrl、Alt)状态的参数,以便在程序中捕捉用户在操作时是否按动了这些控制键。3 个控制键分别用 3 个二进制位来表示,排列顺序如图 6-2 所示。

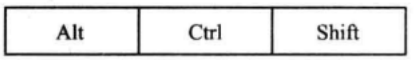


图 6-2 Shift 参数意义

如某一位为 0,则表示对应的键没有按下,为 1 则表示按下了该键。表 6-2 表示了 Shift 参数的可能值。

表 6-2 Shift 参数值及含义

十进制数	二进制数	VB 常数	含义
0	000		3 个键均未按下
1	001	vbShiftMask	按下 Shift 键
2	010	vbCtrlMask	按下 Ctrl 键
3	011	vbCtrlMask + vbShiftMask	同时按下 Ctrl 和 Shift 键
4	100	vbAltMask	按下 Alt 键
5	101	vbAltMask + vbShiftMask	同时按下 Alt 和 Shift 键
6	110	vbAltMask + vbCtrlMask	同时按下 Alt 和 Ctrl 键
7	111	vbAltMask + vbCtrlMask + vbShiftMask	3 个键同时按下

KeyDown 是当一个键被按下时所产生的事件,而 KeyUp 则是松开被按下的键所产生的事件。

【例 6-1】 编写一个程序输出按键的 KeyCode 码。

新建一个窗体名称为“frmExam_6_2”,编写窗体的 KeyDown 事件如下:

```
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
Static i
i = i + 1
If i Mod 10 = 0 Then
    Print Chr(KeyCode); " --"; Hex(KeyCode); "   ";
    Print: Print
Else If KeyCode = 13 Then
    i = 0
    Print: Print
Else
```

```
        Print Chr(KeyCode); " - -"; Hex(KeyCode); "   ";
    End If
    End Sub
```

程序运行后，每按一个键，将输出该键的 KeyCode 码（十六进制）。对于数字键和字母键可以正常输出，对于功能键和其他键，输出的 KeyCode 是正确的，但对应的键并不正确，它们对应了某些小写字母或上挡字符的 ASCII 码。输出结果如图 6-3 所示。

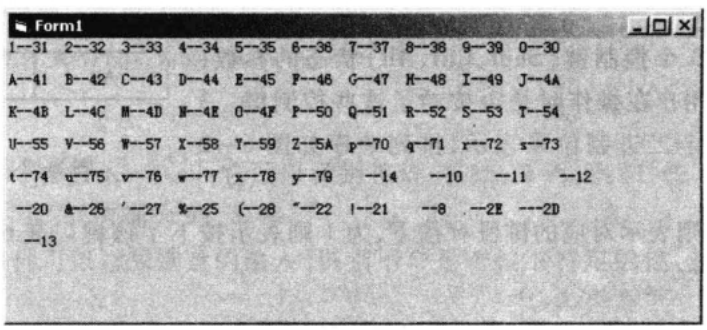


图 6-3　例 6-1 输出结果

① 如果为菜单控件设置了快捷键，那么当按下该键时会自动触发菜单控件的 Click 事件，而不是 KeyDown 事件。

② 当将窗体的 KeyPreview 属性设置为 True 时，窗体将在控件之前识别键盘事件，但对控件来说事件仍会发生。

6.2　鼠标的控制

6.2.1　常用鼠标事件

鼠标与键盘一样，也是用户与程序之间相互交互的主要元素，甚至其重要性超过了键盘。VB 应用程序可以响应多种鼠标事件，除了以前介绍的常用的 Click 事件和 DblClick 事件以外，还可以检测鼠标的位置、判定按下的是左键还是右键、响应鼠标按键与 3 个控制键的组合，如按 Shift 键单击鼠标的常规操作。

要实现这些功能，需要使用以下 3 个鼠标事件。

- MouseDown：按下鼠标按键时触发。
- MouseUp：松开鼠标按键时触发。
- MouseMove：鼠标移动时触发。

这 3 个事件所使用的事件过程模板是非常类似的，除了事件名称有所区别以外，其参数完全相同的。

```
Private Sub 对象名_事件名(Button As Integer, Shift As Integer, X As Single, Y As Single)
...
```

```
End Sub
```

- Button:是一个检测鼠标按键的参数,也是3位二进制数,其0、1、2分别描述鼠标按键的状态。
- Shift:是一个检测3个控制键的参数,其意义与上一节所介绍的相同。
- x,y:表明了鼠标指针的位置,它与接受鼠标事件的对象的坐标系统有关。



当鼠标指针位于无控件的窗体上时,窗体将识别鼠标事件;否则控件将识别鼠标事件。如果按下鼠标不放,则对象将继续识别所有的鼠标事件,即使指针已经离开了对象也是如此,直到用户释放鼠标。

6.2.2 鼠标按键的检测

在编写VB应用程序时,不仅要判定用户按下的是哪个鼠标的按键,而且还要知道用户在操作鼠标时是否按下了控制键及什么控制键,这样才能使程序的功能更加完善。

鼠标按键的检测使用了3个鼠标事件过程中的Button参数,它也是用3个位域来对应鼠标的按键的:0位-对应鼠标左键,1位-对应鼠标右键,2位-对应鼠标中间键(如有的话)。表6-3列出了常用鼠标按键组合值。

表6-3 常用的鼠标按键组合值

十进制数	二进制数	VB常数	含义
1	001	vbLeftButton	按下鼠标左键
2	010	vbRightButton	按下鼠标右键
3	011	vbRightButton + vbLeftButton	同时按下左键和右键



只有MouseMove事件才可以识别是否同时按下鼠标左右键,其他事件不可识别。除了这几个值以外,还有其他4个值,由于均与鼠标中间键有关,对于双键鼠标无法使用。

【例6-2】 利用鼠标事件检测Shift、Ctrl和Alt键的状态。程序运行时按住Shift、Ctrl和Alt三个键中的某个或组合,单击窗体时,会在窗体上显示相应的信息。

新建一个名为frmExam_6_3的窗体,在窗体MouseDown事件中输入以下代码:

```
Dim ShiftTest As Integer
Private Sub Form_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
  ShiftTest = Shift
  Select Case ShiftTest
    Case 1
      Print "You pressed the Shift key."
    Case 2
      Print "You pressed the Ctrl key."
    Case 3
      Print "You pressed the Alt key."
    Case 4
      Print "You pressed both Shift and Ctrl."
    Case 5
```

```
            Print "You pressed both Shift and Alt."
        Case 6
            Print "You pressed both Ctrl and Alt."
        Case 7
            Print "You pressed Shift, Ctrl, and Alt."
    End Select
End Sub
```

【例6-3】 MouseMove 事件与 MouseUp 事件的应用——涂鸦程序。

要求：在按下鼠标左键时开始信手画线，松开鼠标时将停止画线（类似于 Windows 中自带画图程序中的铅笔功能）。

建立窗体 frmExam_6_4，并在不同鼠标事件过程中输入以下代码：

```
Private blndrow As Boolean

Private Sub Form_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
blndrow = True
CurrentX = X
CurrentY = Y
End Sub

Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
If blndrow Then Line -(X, Y)
End Sub

Private Sub Form_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
blndrow = False
End Sub
```

程序中首先在窗体的声明部分声明了一个窗体级变量 blndrow，用于控制是否画线；在 MouseDown 事件中通知应用程序开始画线并将画线的起始位置调整到鼠标当前位置处；在 MouseUp 事件过程中则停止画图（将 blndrow 变量置为 false）；而在 MouseMove 事件过程中则根据 blndrow 变量的状态决定是否画线。程序运行结果如示意图 6-4 所示。



图 6-4　例 6-3 程序运行示意图

6.2.3 鼠标与键盘的结合

在 Windows 应用程序中，有些操作是需要键盘与鼠标相结合的，例如：在资源管理器中选择多个连续的文件时就必须使用 Shift 键加单击，而在 Word 文字处理软件中如要任意地选择文本，就可以通过 Ctrl 加鼠标拖动实现。在 VB 应用程序编制中，利用鼠标事件也可以非常方便地实现这些操作。

要实现鼠标与键盘相结合的编程，关键是不仅要能够捕捉到鼠标的操作，还要能捕捉到键盘的按键操作。在前面已经介绍了鼠标事件过程的 3 个参数，其中之一就是 Shift 参数，该参数在上一节的键盘事件过程中也有，而且使用方法及意义也是相同的，具体含义请参见表 6-2。

【例 6-4】 窗体上有一幅图片，指向图片按 Shift 键并单击，使图片移到窗体的正中间。

先建立一个名为“frmExam_6_5”的窗体，然后在窗体的任意位置上放置一个图片框并半截一个图片，对图片框的 MouseDown 事件过程编写如下程序：

```
Private Sub Picture1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 1 And Shift = 1 Then
  Picture1.Move frmExam_6_5.Width / 2 - Picture1.Width / 2, _
    frmExam_6_5.Height / 2 - Picture1.Height
End If
End Sub
```

该代码中的“Button = 1”判断是否按下的是鼠标左键；而“Shift = 1”则判断是否按下了 Shift 键，如果两个条件均满足，则调用图片框的 Move 方法，将图片移到窗体的正中间。程序的运行结果如图 6-5 所示。

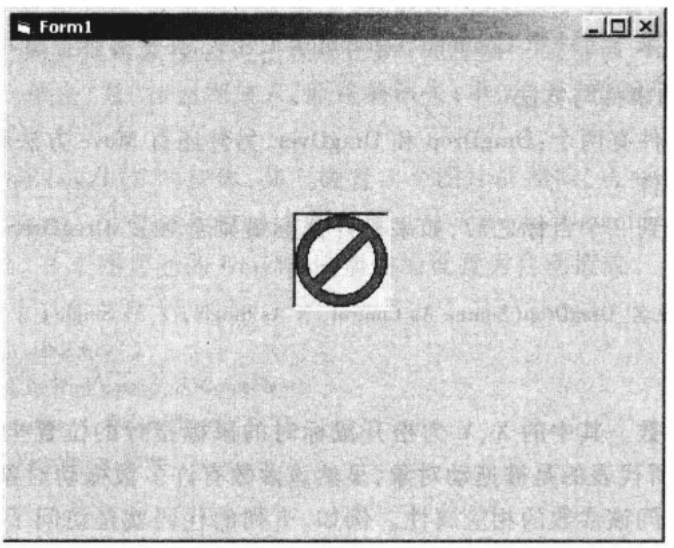


图 6-5　例 6-4 程序运行示意图

*6.3 拖　放

所谓“拖放”是指用鼠标将一个对象从一个地方拖(Dragging)到另一个地方再放下(Dropping)。

拖放操作在 Windows 应用程序中也经常使用,因此 VB 也提供了支持拖放的编程机制。

如图 6-1 所示,案例四就是一个典型的拖放操作,一般将原来位置上的对象称为源对象,而拖放到目的地位置的对象称为目标对象。除了菜单、定时器和通用对话框外,其他控件均可在程序运行期间被拖放。

6.3.1　拖放概述

1. 与拖放相关的属性

凡是可以拖放的对象都有两个与拖放相关的属性:DragMode 与 DragIcon。

(1) DragMode 属性

DragMode 属性用来设置对象的拖放模式,有两种选择:一是自动拖放(1 - Automatic),二是手工拖放(0 - Manual)。所谓自动拖放是指程序运行后该对象可以被自由拖动,而手工拖放则是指程序运行后该对象是固定的,要实现拖放必须使用该对象的 Drag 方法。

DragMode 尽管也只有两种取值,但不是逻辑值,只是一个标志。

如果将一个对象的 DragMode 属性设置为 1,则该对象将不再接收 Click 事件和 MouseDown 事件。

与其他属性一样,该属性既可以在设计时在属性窗口设置,又可以在运行时通过代码设置。

(2) DragIcon 属性

对象在拖动时,移动的并不是对象本身,而是代表对象的一个图标,当松开鼠标放下时才恢复成原来的对象。默认时对象移动时是一个虚框,如果设置了该对象的 DragIcon 属性(该属性包含有指定的图片或图标),在对象移动时就不再是虚框而是指定的图片或图标。

在 VB 的安装目录下有一个 Comman/Ggraphics 文件夹,其中有许多图片或图标可供使用。

2. 与拖放相关的事件与方法

与拖放相关的事件有两个:DragDrop 和 DragOver,另外还有 Move 方法和 Drag 方法。

(1) DragDrop 事件

当把一个对象拖到一个目标之后,如果松开鼠标键则会触发 DragDrop 事件,该事件的格式如下:

```
Private Sub 对象名_DragDrop(Source As Control, X As Single, Y As Single)
...
End Sub
```

该事件有 3 个参数。其中的 X、Y 为松开鼠标时的鼠标指针的位置坐标;而 Source 是一个 Control 类型的参数,所代表的是被拖动对象,显然该参数有许多被拖动对象的属性,在事件过程中可以根据需要来访问该参数的相应属性。例如,下列的代码就是访问了该参数的 Name 属性(对象名是否为 Folder):

```
If Source. Name = "Folder" Then …
```

(2) DragOver 事件

当拖动对象越过另外某一控件对象时,该控件对象将会产生 DragOver 事件。可使用该事件对鼠标指针在一个有效目标上的进入、离开或停顿等进行监控。鼠标指针的位置决定接收该事

件的目标对象。该事件过程的格式与 DragDrop 是类似的，其参数除了具有 DragDrop 过程的 3 个参数以外，还多了一个 State 参数，用于决定鼠标指针与目标控件对象之间的相互位置关系：

- 0 - 鼠标指针正进入目标对象的区域。
- 1 - 鼠标指针正离开目标对象的区域。
- 2 - 鼠标指针正处于目标对象的区域之内。

(3) Drag 方法

由于 Move 方法与一般 Move 方法是相同的，这里主要介绍 Drag 方法。

Drag 方法用于除了 Line、Menu、Shape、Timer 或 CommonDialog 控件之外的任何控件的开始、结束或取消拖动操作。其格式为：

对象名.Drag 动作

该方法有一个参数：动作。它有 3 种不同的取值：

- 0 - 取消拖动操作。
- 1 - 开始拖动指定的对象。
- 2 - 结束对指定对象的拖动并发出一个 DragDrop 事件。

只有当对象的 DragMode 属性设置为手工(0)时，才需要使用 Drag 方法控制拖放操作。但是，也可以对 DragMode 属性设置为自动(1)的对象使用 Drag。

6.3.2 拖放控制

从前面的介绍可知，拖放可分为自动拖放与手工拖放，本小节通过两个实例来说明拖放的操作方法。

【例 6-5】 自动拖放示例：编写一个类似于“回收站”的程序。用适当的图标作为回收站，程序运行后，把窗体上的其他对象拖放到回收站上后，即弹出一信息框，询问“是否确实需要将该对象放入回收站?”，单击“是”按钮则放入，原对象消失；单击“否”按钮则操作取消；双击窗体则恢复。

新建一个名为 frmExam_6_6 的窗体，其上放置 3 个图片框控件，从 Comman/Ggraphics 文件夹中选取适当的图标装入。另外再建立一个图片框，从 Comman/Ggraphics 中选取回收站(空)的图标如图 6-6 所示。3 个图片框的 DragMode 属性均设置为自动拖放。

相关程序代码如下：

```
Private Sub Form_DblClick( )
For Each Control In frmExam_6_6. Controls
    If Control. Visible = False Then
        Control. Visible = True
    End If
Next
Picture1. Picture = LoadPicture( " F: \Program Files\Microsoft Visual Studio\Common\ _
                    Graphics\Icons\Win95\waste. ico" )
End Sub
```

```
Private Sub Picture1_DragDrop( Source As Control, X As Single, Y As Single)
Dim ans
```

```
ans = MsgBox("是否确实将该对象放入回收站?", vbYesNo)
If ans = vbYes Then
  Source.Visible = False
  Picture1.Picture = LoadPicture("F:\Program Files\Microsoft Visual Studio\Common\ _
                    Graphics\Icons\Win95\recyfull.ico")
End If
End Sub
```

程序运行后的结果如图6-7所示。

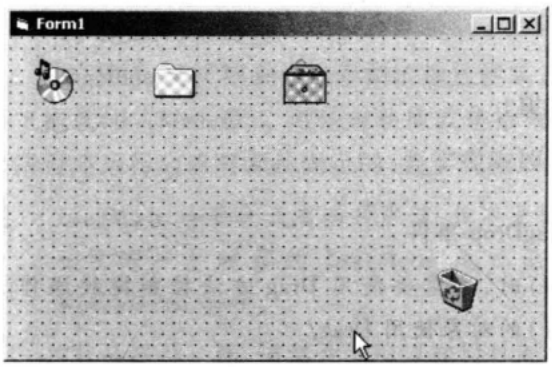


图6-6　模拟回收站设计图

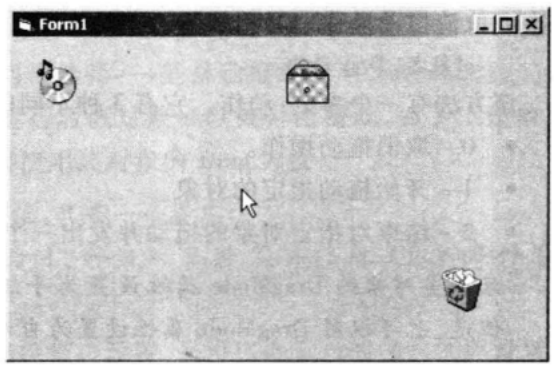


图6-7　程序运行示意图

【例6-6】　手工拖放示例：设计一个如图6-8所示的应用程序。左边的图片框采用手工拖放模式，标签采用自动拖放模式。图片框只能使用左键拖动，拖到右边的图像框后图片框消失，且拖动过程中图片框中的图片作为拖动图标使用。标签拖到图像框后显示“ERROR!”。

新建一个名为frmExam_6_7的窗体，放置相关控制并进行属性设置。编写相关对象的事件过程如下：

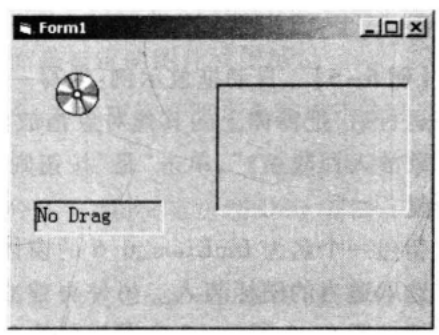


图6-8　手工拖放示意图

```
Private Sub picture1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 1 Then                              '判断是否是左键
    Picture1.Drag 1                             '手工启动拖放
    Picture1.DragIcon = Picture1.Picture        '设置拖动图标
End If
End Sub
```

```
Private Sub image1_DragDrop(Source As Control, X As Single, Y As Single)
If TypeOf Source Is Picture Then                '判断拖动源是否为图片框
  Image1.Picture = Source.Picture               '图像框中装入源的图片
    Source.Visible = False                      '源图片消失
Else
```

```
    MsgBox ("ERROR!")
End If
End Sub
```

6.4 案例四设计

利用本章所学习的知识可以进行案例四的设计。本案例实际上也是一个鼠标拖放的实例。

根据案例的要求进行界面设计,其中文件夹的拖放模式设为自动。文件夹的名称设为"folder",其拖动图标设置为一个文件图标;文件柜的名称设为"cabinet",其图标使用两个:一个是抽屉关闭状的(files03a. ico),另一个是抽屉打开状的(files03b. ico)。案例中所使用的图标在"Common/Graphics"下均可以找到。所设计的界面如图 6-9 所示,窗体名为"frmExam_6_1"。

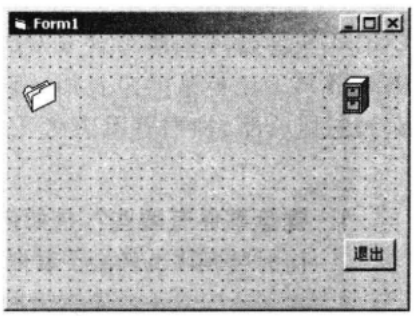


图 6-9 案例四界面设计

代码设计要点:

对文件柜要同时考虑 DragDrop 事件和 DragOver 事件,DragDrop 事件的任务:一是将文件柜图标恢复成关闭状(表示文件已经接收),其二是在标签中显示相应的"文件已经接收"的内容。而 DragOver 事件的任务是:

① 判断拖动源是不是文件夹,如是则将文件柜图标换成抽屉打开的图标。

② 如果拖动离开了文件柜则再将文件柜图标换成抽屉关闭的图标(通过 State 参数可以判定)。

为了能在拖动过程中判定鼠标的位置,从而根据不同的位置在标签中显示不同的信息,需要编写标签(label1)的 DragOver 事件过程,根据其 State 状态,分别设置不同的提示信息。完整代码如下:

```
Private Sub cabinet_DragDrop(Source As Control, X As Single, Y As Single)
If Source. Tag = "folder" Then
    cabinet. Picture = LoadPicture("F:\Program Files\Microsoft Visual Studio\Common\ _
                       Graphics\Icons\Office\files03a. ico")
    Label1. Caption = "文件已经接收!"
End If
End Sub
```

```
Private Sub cabinet_DragOver(Source As Control, X As Single, Y As Single, State As Integer)
If Source. Tag = "folder" Then
   cabinet. Picture = LoadPicture("F:\Program Files\Microsoft Visual Studio\Common\ _
                      Graphics\Icons\Office\files03b. ico")
End If
If State = 1 Then
   cabinet. Picture = LoadPicture("F:\Program Files\Microsoft Visual Studio\Common\ _
                      Graphics\Icons\Office\files03a. ico")
End If
End Sub
```

```
Private Sub Command1_Click( )
End
End Sub
```

```
Private Sub Label1_DragOver(Source As Control, X As Single, Y As Single, State As Integer)
Select Case State
    Case 0
        Label1. Caption = "正在进入目标区域!!!"
    Case 1
        Label1. Caption = "离开了目标区域!!!"
    Case 2
        Label1. Caption = "已经位于目标区域!!!"
End Select
End Sub
```

习 题 6

1. 键盘事件有哪些? 是如何引发的?

2. 何时使用 KeyPress 事件? 何时使用 KeyDown 和 KeyUp 事件?

3. 在 KeyDown 事件中如何检测 Ctrl 和 F3 是否同时被按下?

4. 计算机键盘上的"4"键的上档字符是"$",当同时按下 Shift 键和"4"键时,KeyPress 事件发生了几次? 在过程中的 KeyAscii 的值是多少?

5. 确保文本框中输入的全部是数字的最佳方法是什么?

6. 窗体的 KeyPreview 属性的作用是什么?

7. 如果在 KeyDown 事件过程中将 KeyCode 设置为 0,KeyPress 的 KeyAscii 参数会不会受影响?

8. 请说明 KeyCode 与 KeyAscii 的区别?

9. 在窗体上放置两个图片框控件 picture1 和 picture2,在 picture1 中装入一幅图片。程序运行时拖动 picture1(拖动图标就是图片本身)刚进入、经过、离开 picture2 时,分别为 picture2 设置 3 种不同的前景色。

10. 编写一个程序,当同时按下 Alt、Shift 和 F6 键时,在窗体上显示"Good Bye!",并终止程序的运行。

11. 在窗体上设计一个文本框、一个图片框和一个命令按钮。编写程序使得当光标位于不同的控件或窗体上时,鼠标具有不同的指针,此时如果按下鼠标右键则显示相应的信息。

12. 编写程序,在窗体上画圆。要求:按着鼠标右键移动可画圆,否则不行。VB 中画圆的命令为:"Circle(X,Y),R",即以 X、Y 为圆心,以 R 为半径画一个圆。

第7章　菜单与工具栏

学习目标

掌握 MDI 窗体的创建方法以及在 MDI 窗体上熟练地使用 VB 提供的菜单设计器进行用户下拉式菜单及弹出式菜单的设计；并能利用 VB 功能扩展方法（使用 ActiveX 控件方法）进行工具栏的设计，并在应用程序中将工具栏与菜单结合设计用户界面。

友好的用户界面是衡量一个应用软件开发水平的重要标志，而菜单和工具栏是用户界面的重要组成部分。菜单与工具栏的任务是将应用程序所能完成的任务有机地组织起来以方便用户使用。在 Windows 环境下，几乎所有的软件都有菜单与工具栏，而且这些菜单与工具栏从构成、组织到操作方法甚至快捷键都非常相似。因此，使用 VB 来开发应用程序时，都必须进行菜单与工具栏的设计。

【案例五】 学生成绩管理系统的菜单及工具栏。

在信息管理系统软件的界面设计中，经常要进行菜单及工具栏的设计。通过菜单及工具栏将信息管理系统所需的窗体进行集成，即通过菜单及工具栏进行调用，从而达到信息管理系统界面设计的目的。本案例就是在 MDI 窗体上利用 VB 的菜单设计器和 VB 功能扩展方法（使用 ActiveX 控件方法）进行菜单及工具栏的设计，制作出如图 7-1 所示的菜单及工具栏界面。在该界面中通过单击菜单或工具栏上相应的按钮调出第 5 章中建立的窗体 frmexam_5_8.frm、frmexam_5_9.frm、frmexam_5_5.frm。

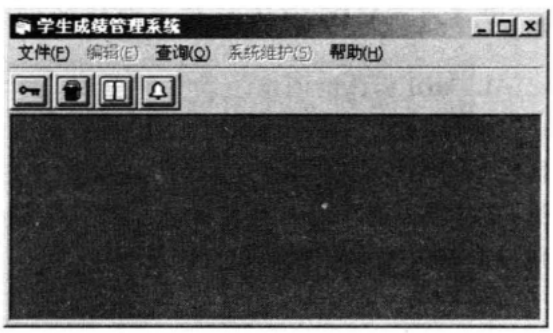


图 7-1　在 MDI 窗体上制作菜单及工具栏

7.1　多文档界面

多文档界面（Multiple Document Interface，MDI）一般是大型应用程序所具有的界面（如 Word、Excel 等）。MDI 是一个容器窗体，它就像是整个应用系统的背景，如果移动容器窗体，则包含在容器窗体的窗体也随着移动；关闭容器窗体时，容器内的所有窗体也一起关闭。该容器窗体被称为 MDI 父窗体，包含在父窗体内的其他窗体称为 MDI 子窗体。

通常，信息管理系统由一个 MDI 父窗体和多个 MDI 子窗体构成，它允许用户同时显示多个文档，每个文档显示在一个子窗体中。每个子窗体均被限制在父窗体中。MDI 应用系统中父窗体和子窗体都有自己的特点。

1. MDI 父窗体的特点

① 一个应用系统只能有一个父窗体。

② 不能使用 Print 方法或任何图形方法在父窗体中显示信息。

③ 父窗体最小化时,父窗体和所有的子窗体都用 Windows 任务栏上的一个图标表示;当父窗体还原时,所有子窗体按照它们在最小化之前的相同布局还原。

④ 父窗体和子窗体都可以有自己的菜单,但在窗体加载时,菜单的显示将是不一样的。如果含有菜单的子窗体被加载,则子窗体菜单将取代父窗体的菜单而显示在父窗体的菜单栏中;如果含有菜单的子窗体没有被加载,则父窗体菜单栏中显示自己的菜单。

2. MDI 子窗体的特点

① 当 MDI 子窗体的边框大小可变时,其初始大小及其显示位置由父窗体大小决定,与其在设计时的窗体大小无关;而当 MDI 子窗体的边框大小不可改变时,该子窗体的大小由设计时的 Width 属性和 Height 属性决定。

② 当子窗体最小化时,它的图标显示在父窗体中,而不会出现在任务栏中。

③ 当子窗口最大化时,它只能填满整个父窗口,而不会填满整个屏幕,并且父窗体的标题栏包含父窗体和子窗体的名称,但在任务栏中出现的仅有父窗口。

④ 当一个子窗体最大化时,所有的子窗体都随着最大化。

本节将创建一个如案例五的 MDI 窗体。

7.1.1 MDI 窗体及其子窗体的创建

1. MDI 窗体的创建

选择〖工程〗→〖添加 MDI 窗体〗菜单命令或单击工具栏上“添加窗体”按钮中的“添加 MDI 窗体”,均可出现如图 7-2 所示的“添加 MDI 窗体”对话框,在该对话框中既可以新建一个 MDI 窗体,又可以将现存的 MDI 窗体添加到工程中。当选择“MDI 窗体”并单击“打开”按钮后,即可创建一个 MDI 窗体。

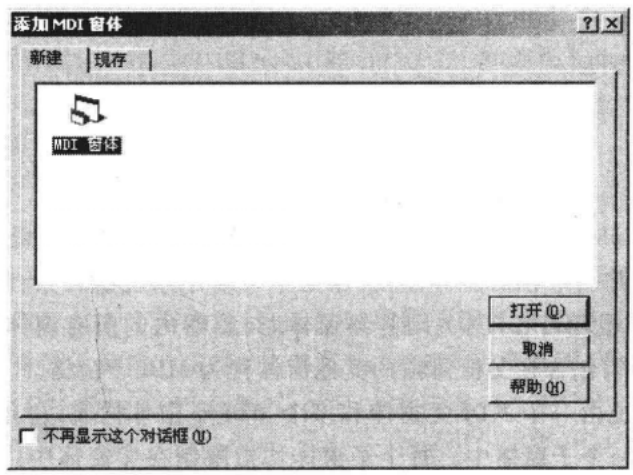


图 7-2 “添加 MDI 窗体”对话框

一个应用程序只能有一个 MDI 窗体，如果在工程中已经存在了一个 MDI 窗体，则“添加 MDI 窗体”命令将不可用。

在属性窗口中修改该 MDI 窗体的名称(“MDI”)和标题(“学生成绩管理系统”)。

MDI 窗体是一个特殊的窗体，在该窗体上可以设计菜单，也可以放置工具栏。但大部分控件都不能直接放置在 MDI 窗体上，只有具有 Align 属性(如 PictureBox 控件)或具有不可见界面的控件(如 Timer 控件)才能出现在 MDI 窗体上。

2. MDI 子窗体的创建

当 MDI 父窗体创建完成后，即可创建其子窗体。实际上子窗体与一般普通的窗体之间的差别在于：子窗体的 MDIChild 属性为 True。因此，子窗体可以是一个新建的窗体，也可以是一个已经存在的窗体。只要将已经存在的窗体的 MDIChild 属性改为 True 即可变成子窗体。子窗体的设计过程与普通的窗体是一样的，可以添加各种控件、设置各控件的属性和编写各控件的事件代码等。

【例 7-1】 在加载 MDI 窗体的工程中将第 5 章中建立的普通窗体(frmexam_5_8. frm、frmexam_ 5_81. frm frmexam_5_82. frm、frmexam_5_9. frm、frmexam_5_5. frm)添加到工程中，并将其设置为 MDI 子窗体。

读者可以将第 5 章已经建立的窗体(frmexam_5_8. frm、frmexam_5_81. frm、frmexam_5_82. frm、frmexam_5_9. frm、frmexam_5_5. frm)添加到工程中，并将添加到项目中的各窗体的 MDIChild 属性均改为 True，即各窗体被变成 MDI 子窗体，然后再将 main1. bas 添加到模块中。添加后的 MDI 窗体及子窗体如图 7-3 所示。

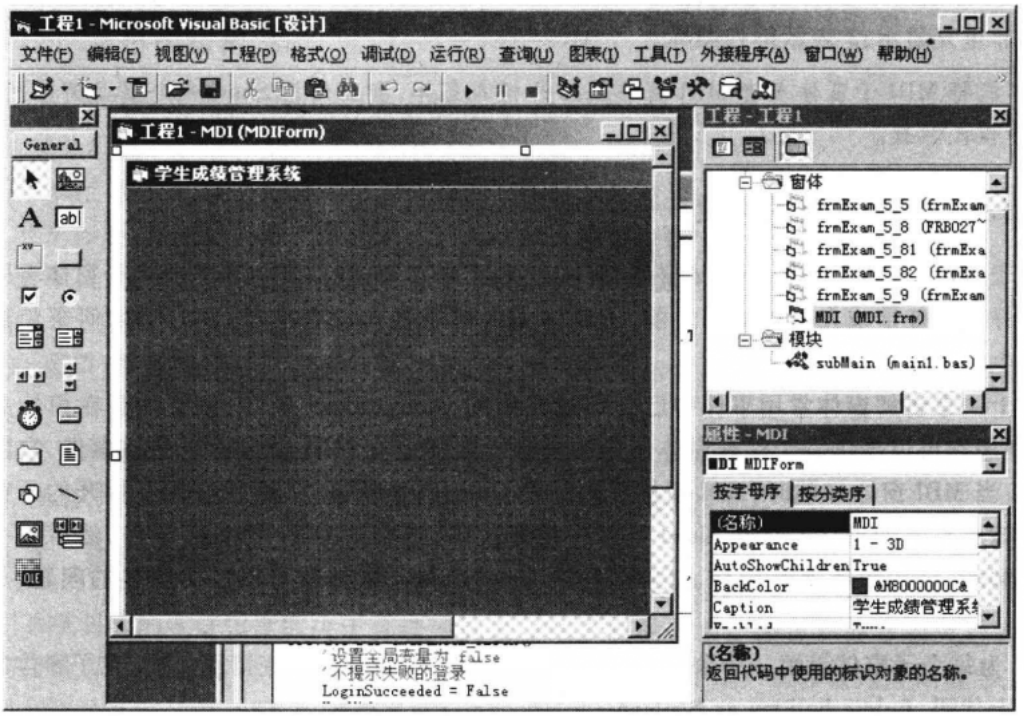


图 7-3 添加 MDI 窗体并将现存的窗体添加到项目中并将其设置为子窗体

对于一个应用程序而言，除了父窗体以外，其他的窗体并不一定非要是子窗体，它们也可以

是标准的非 MDI 子窗体。

通过查看每个窗体的 MDIChild 属性或检查工程资源管理器，可以确定窗体是否是一个 MDI 子窗体。在工程资源管理器中，MDI 父窗体、子窗体及普通窗体的显示图标是不同的。代表普通窗体图标，代表 MDI 子窗体图标，代表 MDI 窗体图标。

7.1.2 MDI 窗体的显示

MDI 窗体及其子窗体的运行方法与普通窗体的运行方法类似。由于在 MDI 窗体上一般都要放菜单和工具栏，因此，当应用程序具有 MDI 窗体时，总是将 MDI 窗体设置为较早启动的窗体（一般在 Flash 窗体和登录窗体之后）。父窗体加载显示时不会自动加载显示子窗体，一般需要利用代码加载显示子窗体，子窗体的加载与显示代码均放置在 MDI 窗体中相应菜单或工具栏的事件过程中。如果在父窗体没有显示的情况下需要调用显示子窗体时，父窗体会和子窗体一起被调用显示。

如果希望在加载显示父窗体的同进显示子窗体，需要设置 MDI 窗体的 AutoShowChildren 属性值为 True，则在加载显示 MDI 窗体时会自动显示所有的子窗体。

MDI 窗体中同时可以有许多的子窗体显示，但是每次只能有一个子窗体是激活的（即该窗体具有输入焦点）。通过 MDI 窗体的 ActiveForm 和 ActiveControl 属性，可以查到当前活动窗体拥有输入焦点的控件。当前激活的窗体名称可以用 Me 关键词来代替。

当一个应用系统的子窗体比较多时，子窗体的显示可以有多种排列，其实现可以用 Arrange 方法，该方法有 4 个参数，取值分别为 0～3，其中 0 表示层叠显示；1 表示水平非重复显示；2 表示垂直非重复显示；3 表示排列图标。



不能将 MDI 子窗体或者 MDI 窗体显示为模式窗体，即在使用 Show 命令显示窗体时不能加模式参数。

7.1.3 MDI 窗体的卸载

由于 MDI 子窗体属于 MDI 父窗体，所以当关闭 MDI 窗体时，其包含的所有子窗体将同时关闭。这样有可能造成在窗体关闭之前，子窗体中的信息根本没有来得及保存，如何避免这种情况的发生呢？

在上一章讲解窗体常用事件中已介绍过窗体的 QueryUnload 事件，该事件会在窗体关闭时触发，利用它可以提供一个保存窗体信息的机会。实际上在 MDI 窗体中也有该事件，它的触发过程是：当 MDI 窗体卸载时，MDI 窗体首先触发 QueryUnload 事件，然后每一个打开的子窗体也都触发该事件。如果在 QueryUnload 事件中没有代码，则取消 QueryUnload 事件，然后每一个子窗体被卸载，最后卸载 MDI 窗体。因此，利用 QueryUnload 事件就可以解决上述的问题，具体思路是：

① 为每个子窗体设置一个公用变量以记录该窗体中的信息改变情况。如，在子窗体的声明部分加上代码：Public bolFlag as Boolean。

② 根据窗体上的信息变化情况设置该公用变量的值。当某些控件的信息改变后，在相关的事件过程中（如文本框的 Change 事件）置 bolFlag 为 True，而当窗体上的信息保存后（如单击“保

存”按钮)则在保存的事件过程中将该标志置为 False。

③ 在 MDI 窗体的 QueryUnload 事件中判断 bolFlag 的值以确定是否需要提醒用户保存数据。例如可以在 QueryUnload 事件输入如下代码:

```
Private Sub MDIForm_QueryUnload(Cancel As Integer, UnloadMode As Integer)
    Dim intYn As Integer
    If ActiveForm.bolFlag Then
    intYn = MsgBox("你还没有保存数据,需要保存吗?", vbYesNo + vbExclamation, "询问")
        If intYn = 6 Then          '单击 Yes 按钮
            Cancel = 1             '禁止卸载窗件,便于用户保存数据
        End If
      End If
End Sub
```

在该代码中 ActiveForm 是 MDI 窗体的一个属性,表示当前活动的子窗口。与此类似的还有一个 ActiveControl 属性可以返回活动窗体上的活动控件(即具有焦点的控件),因此利用该属性可以很方便地确定窗体上哪个控件具有焦点,以便进行进一步处理。

当需要使用 ActiveForm 属性时,至少应该有一个子窗体可见,否则会返回一个错误。

7.2 菜单的规划与设计

7.2.1 菜单的结构

Windows 环境下的各种应用软件通常都有自己的菜单,而且菜单的结构都相同。菜单系统一般由 4 部分组成。

- 菜单栏:指位于窗口标题栏下方包含若干个菜单标题的水平矩形条状区域。
- 菜单标题:指菜单栏中菜单的名称。每个菜单标题标识了一个下拉列表。
- 菜单项:包含于菜单之中的各个选项,当选中某个菜单项后,可能会执行一条命令,也可能是执行一个过程或函数,甚至可能弹出一个子菜单。
- 子菜单:是指激活一个带有“▶”标记的菜单选项后打开的列表,它由一系列菜单项构成。

在每个菜单标题或菜单项右侧均有相应的访问键,有些菜单项还有快捷键,有关访问键和快捷键的概念、操作方法与 Windows 中菜单的使用方法是一致的。

7.2.2 菜单的规划

应用系统界面的友好程度取决于合理地规划应用系统的菜单。因此对菜单系统进行精心的规划与设计,不仅有助于准确实现应用程序的功能,而且有助于用户操作使用应用程序。

在进行菜单系统的规划时,一般应考虑如下原则:

(1) 面向用户

要从使用者的角度出发,按照用户所要执行的任务流程来组织菜单系统,而不要按应用程

序的层次组织菜单系统。

(2) 菜单名称应通俗易懂

在给每个菜单或菜单项进行命名时,要尽量避免使用那些过于专业的甚至似是而非的名词,而应该使用比较通俗易懂又能体现其功能的日常用语。

(3) 合理组织菜单项

一般情况下,对菜单项目的组织要按照菜单项的使用频率或者菜单项使用的逻辑顺序,这样有助于提高操作的效率,同时使用起来也得心应手。

(4) 设置合适的分隔线

按照菜单项之间的逻辑关系,将内容相近的菜单项合并在一个逻辑组中,并在每个逻辑组之间放置分隔线。

(5) 控制菜单项的数目

菜单的项目个数不能超过一个屏幕。如果项目过多,超过一个屏幕,可以考虑将其中某些菜单项设置成子菜单。

(6) 设置合适的访问键和快捷键

功能完善的菜单系统既可以使用鼠标,也可以使用键盘进行操作,对于那些需要经常使用的菜单项,还应该为其设置快捷键以提高操作效率。在指定访问键和快捷键时,要注意遵守Windows 应用程序的习惯。例如,Alt + F 键为"文件"菜单的访问键,用 Ctrl + P 键为"打印"菜单项的快捷键。

例如,对于本书中所使用的"学生成绩管理系统"的菜单体系结构如表 7-1 所示。

表 7-1　"学生成绩管理系统"菜单体系结构

文件(F)	编辑(E)	查询(Q)	系统维护(S)	帮助(H)
注册(L)	成绩编辑(C) Ctrl + J	按班级(B) Ctrl + I	课程设置(K)	内容 F1
退出(X)	补考录入(B) Ctrl + B	按个人(G) Ctrl + G	学生档案(U)	关于…
		通用查询(T) Ctrl + T	班级编码(B)	
			课程编码(C)	
			专业目录(Z)	
			用户管理(M)	

7.2.3　下拉式菜单的设计

利用 VB 菜单编辑器不仅可以设计一个新的菜单,而且可以对已有菜单进行修改。在整个菜单的设计过程中,通过全交互方式即可完成菜单的制作,几乎不需要编程。

【例 7-2】　在【例 7-1】的 MDI 窗体上,利用菜单编辑器完成表 7-1 所示菜单的制作。

1. 菜单编辑器

要使用"菜单编辑器",必须首先选择一个拟放置菜单的窗体。本例选择在【例 7-1】的 MDI 窗体上放置菜单,该 MDI 窗体为应用程序的主窗体,然后选择〖工具〗→〖菜单编辑器〗菜单命令或单击"标准"工具栏上的"菜单编辑器"按钮均可打开"菜单编辑器"的对话框,如图 7-4 所示。

在 VB 中，每个菜单标题或菜单项均被看成在“菜单编辑器”中所定义的一个菜单控件，它们与命令按钮这样的控件没有什么区别，也可定义其外观或行为的各种属性，如：Name、Caption、Enabled、Visible 和 Checked 等。菜单控件的事件比较简单，只有一个 Click 事件，当用键盘或鼠标选中菜单项时，将触发相应的 Click 事件。

菜单控件的大部分属性都可以通过如图 7-4 所示的“菜单编辑器”对话框来设置。当然，在运行时也可以对某些属性进行重新设置以达到预定的功能。

从图 7-4 可以看出，菜单编辑器大体可分为 3 个区域：

(1) 属性设置区

属性设置区位于“菜单编辑器”中间一排按钮上方的部分，这一部分主要是对菜单控件的各种属性进行设置，如标题、名称、索引号、是否有效和是否可见等，其中“名称”是设计菜单程序的唯一标识。

(2) 菜单控件列表框

在菜单控件列表框中列出了当前窗体的所有菜单控件，它位于“菜单编辑器”的下部。VB 将位于不同位置的菜单赋予了不同的层次，各个层次通过在列表框中进行适当的缩进来区分。第一层是菜单标题控件，第二层是菜单选项控件，第三层是子菜单选项控件，依此类推。如图 7-5所示即是在【例 7-1】的 MDI 窗体上，利用菜单编辑器已经完成表 7-1 的“学生成绩管理系统”的菜单的编辑状态。

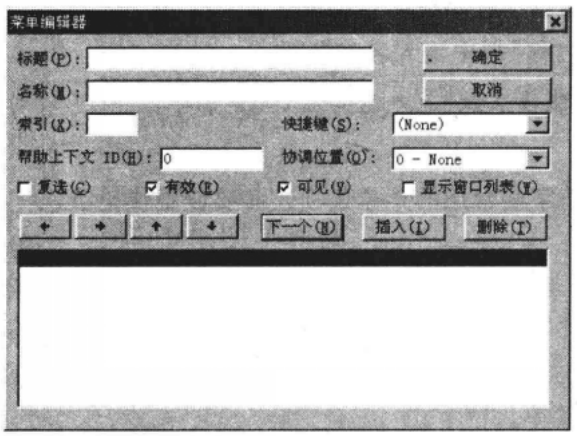


图 7-4 “菜单编辑器”对话框



图 7-5 “学生成绩管理系统”菜单编辑状态

(3) 菜单控件编辑按钮

它位于“菜单编辑器”的中间，包括 4 个方向箭头和 3 个编辑按钮，利用这几个按钮可以改变菜单控件的层次(菜单标题控件、菜单选项控件和子菜单项控件)，插入或删除菜单项。

2. 菜单的创建

在“菜单编辑器”中要创建一个菜单实际上是非常简单的，主要步骤如下：

① 打开要创建菜单的窗体(本例是在【例 7-1】的 MDI 窗体上建立菜单)，也可以新建一个窗体。

② 打开如图7-4所示的“菜单编辑器”对话框。

③ 在图7-4的“标题”文本框中输入希望在菜单上显示的文本，如“文件”。该标题实际上是菜单控件的Caption属性。

④ 在“名称”框中输入该菜单控件的名称，用于在代码中引用，如mnuFile。命名时一般要体现菜单项所在的层次。例如，“文件”菜单下的“注册”选项可命名为mnuFile_Login。

⑤ 通过“➡”和“⬅”按钮来改变菜单项的层次，“➡”按钮将菜单项降低一层，即如果原来是菜单标题的话，降低后则变为菜单选项，降低后在控件前出现“…”符号。每单击一次降低一层。“⬅”按钮的作用正好与此相反。“⬆”和“⬇”两个按钮用于改变菜单控件的顺序。

⑥ 如果需要，还可以设置菜单控件的其他属性。

⑦ 当一个菜单控件创建完成后，可以通过单击“下一个”按钮再创建下一个菜单控件。

☞ 由于“下一个”按钮为默认按钮，因此直接按Enter键也可进入下一项菜单的输入。

⑧ 重复③～⑦步，直到将所有的菜单全部创建完成，单击“确定”按钮关闭“菜单编辑器”对话框。

3. 菜单的分组

菜单的分组是指将具有一定逻辑关系的菜单选项分成一个组（如常见的剪切、复制与粘贴通常在一个组），各组之间要用分隔线隔开。创建分隔线既可在创建菜单时，也可在现有菜单中添加。在现有菜单中添加的方法是：首先需要选择添加分隔线的菜单项目，单击“插入”按钮，在“标题”框中输入连字符“-”（注意是半角），在“名称”框中输入名称即可。

❗ 虽然分隔线也算是一个菜单控件，但它既不能被选取，也不能响应Click事件。

4. 菜单访问键与快捷键

对菜单标题或菜单选项可以设置访问键，对于某些常用的菜单选项还需要设置快捷键。

（1）访问键的设置

在菜单编辑器中选择要加访问键的菜单控件，在其“标题”框文本前面或后面输入前导字符“&”，再输入要作为访问键的字母。例如，在“文件”菜单中想使用F键作为访问键，可在其“标题”框输入“文件(&F)”或“(&F)文件”的文字。这样当菜单运行时，它就会变成“文件(F)”或“(F)文件”的式样，用快捷键Alt+F即可打开该菜单。

❗ 在同一个菜单标题下，不能使用相同的访问键，否则这个访问键不会起作用。

（2）快捷键的设置

快捷键用于在不打开菜单的情况下快速执行命令，一般只对那些使用频繁的菜单选项设置快捷键，而且快捷键在整个菜单系统中应该是唯一的，不能出现相同的快捷键。

快捷键的设置方法也是首先选择要设置快捷键的控件（一般是菜单选项），然后在“快捷键”框中选取一个快捷键（通常都是各种组合键）。

当快捷键设置好后，在下部的控件列表框中相应控件的右侧就会出现所选择的快捷键名称。

☞ 如要删除一个已经设置好的快捷键，请在“快捷键”框中选择“None”。

快捷键的设置要符合一般的 Windows 软件的习惯，不要与已定义的快捷键发生冲突。例如 Ctrl + C、Ctrl + X、Ctrl + V 已经分别用于读者十分熟悉的复制、剪切与粘贴菜单的快捷键，在应用程序中就不要另外设置成其他菜单的快捷键。

5. 菜单控件事件代码

前已述及，应用程序的所有功能都是通过菜单来实现的，用户一般也是通过使用应用程序的菜单来调用其相应的功能。当一个菜单系统利用以上的方法创建完成后，只能算完成了一半，因为更重要的是要将各个有关的菜单选项与某个窗体联系起来。也就是说，在程序运行时，当选中了某个菜单选项后，能运行指定的窗体以完成预期的功能（由菜单的 Click 事件的代码实现），因此要实现菜单相应的功能必须通过选择菜单选项控件，编写菜单选项控件所对应的 Click 事件代码。

菜单创建完成后，所创建的菜单标题将显示在所在的窗体上。单击某一个菜单标题即出现该菜单标题对应的菜单选项。但请注意，此菜单这时并未运行，仍然处于设计状态，通过单击某一个菜单选项可以打开相应的 Click 事件代码窗口，在此窗口中可编写相应的代码。

要在某个菜单控件的 Click 事件过程中调用某一个窗体，只要在其代码窗口中填入类似于如下形式的命令即可（请参阅第 5 章的有关内容）。

以下是在 MDI 窗体上注册、通用查询、课程设置和退出等菜单所对应的 Click 事件代码。分别调用了第 5 章中 frmexam_5_8. frm、frmexam_5_9. frm、frmexam_5_5. frm 的窗体文件。

```
Private Sub MDIForm_Load( )   'MDI 窗体加载事件
MDI. MNUEDIT. Enabled = False
MDI. MNUALTER. Enabled = False
MDI. MNUFIND. Enabled = False
MDI. MNUHELP. Enabled = False
MDI. MNUEDIT. Enabled = False
MDI. MNUFILE. Enabled = True
End Sub
```

```
Private Sub MNUFILE_EXIT_Click( )        '退出菜单的 Click 事件
TCXT = MsgBox("真的要退出", vbOKCancel, "退出系统")
If TCXT = 1 Then Unload Me
End Sub
```

```
Private Sub MNUFILE_LOGIN_Click( )       '注册菜单的 Click 事件
frmExam_5_8. Show
frmExam_5_9. Hide
frmExam_5_5. Hide
End Sub
```

```
Private Sub MNUALTER_KC_Click( )         '课程设置菜单的 Click 事件
frmExam_5_5. Show
frmExam_5_9. Hide
frmExam_5_8. Hide
End Sub
```

```
Private Sub MNUFIND_UNIVERSE_Click()  '通用查询菜单的 Click 事件
frmExam_5_9. Show
frmExam_5_8. Hide
frmExam_5_5. Hide
End Sub
```

6. 在运行时设置菜单控件

在 Windows 应用软件的菜单系统中，菜单选项的表现形式共有 6 种（直接执行、无效、弹出对话框、选中、单选和子菜单），分别使用不同的方式来表示。例如：暂时无效的菜单选项是灰色的，在菜单选项前加“√”表示选中，等等。另外，有些菜单下的选项是动态的，某些项目有时是可见的，有时又是隐藏的，所有这些都可以在运行时通过对菜单控件进行适当的设置来实现。

（1）增减菜单项

菜单控件的 Visible 属性决定了该菜单在运行时是否可见。利用这个属性，可以在程序运行时动态地增减菜单项。设计时要将所有的菜单项全部设计完成，通过清除菜单编辑器中的“可见”检查框就可以将选中的菜单项隐藏起来。程序运行时如需要该菜单项显示，只要将该菜单项的 Visible 属性设为 True 即可。

有些情况下，不仅菜单项的数目是动态的，而且菜单项的名称也是动态的（如 Word 软件中“文件”菜单下列出的最近打开的文档列表）。显然，用上述方法是无法实现这种功能的，在 VB 中提供了一种称为“菜单控件”数组的特殊数组，利用它可以实现这种要求。关于这一方面的内容，感兴趣的读者请参考其他相关资料或 VB 联机帮助。

（2）在菜单项上使用选中标记

有些菜单项本身是一个开关，打开即执行某项功能，关闭则不执行。如控制应用程序状态栏的显示/关闭。在菜单项“状态栏”前加上选中标记“√”表示要打开状态栏，去掉“√”表示要关闭状态栏。利用菜单控件中的 Checked 属性可以实现这一功能，当 Checked 属性为 True 时表示选中，“√”将出现在菜单项的左侧；当 Checked 属性为 False 时，“√”标记消失，表示关闭某功能。如果在设计时想使某菜单项带上“√”标记，只要选中菜单编辑器中相应菜单项的“复选”框就可以了。

（3）使菜单选项有效或无效

有些菜单命令在执行时是需要一定条件的。如果条件不满足，用户是不能选择的。例如，如果在剪贴板上没有内容，那么“粘贴”命令就不能执行，其在菜单上的表现为灰色。此时不仅对鼠标不做响应，而且键盘的快捷键也同样不起作用。控制菜单控件的 Enabled 属性可以实现此功能。每个菜单控件都有一个 Enabled 属性，当 Enabled 属性为 True 时，该菜单选项有效，否则无效。显然，这种设置只能在程序运行时根据实际情况进行。

例如：在“学生成绩管理系统”中，在“系统登录”窗体中设置一个公用变量 LoginSucceeded，当其值为 True 时表示是系统管理员，将“编辑”与“系统维护”菜单置为有效；否则就是非系统管理员，对非系统管理员不允许其进行数据编辑或维护。在菜单编辑器中，首先将这两个菜单设置为无效，然后在“登录”窗体中“确定”按钮的 Click 事件过程中输入以下代码：

```
Private Sub cmdok_click()
```

```
    '检查正确的密码
    Dim i As Integer
    For i = 1 To 4
    If txtpassword. text = strpass(i) And _
    txtusername. text = strname(i) _
       And txtusername. text < > "" Then
       msgbox "密码正确,欢迎进入!", "登录"
       intuser = i
       me. hide
       Exit For
     End If
  Next i
   If strpass(i) = "system" and strname(i) = "system" Then
     loginsucceeded = true
   Else
     loginsucceeded = false
   End If
   If Not loginsucceeded Then
   msgbox "该用户不是系统管理员", "登录"
       mdi. mnuedit. enabled = false
       mdi. mnualter. enabled = false
       mdi. mnufind. enabled = true
       mdi. mnuhelp. enabled = true
       mdi. mnufile. enabled = true
   Else
       mdi. mnuedit. enabled = true
       mdi. mnualter. enabled = true
       mdi. mnufind. enabled = true
        mdi. mnuhelp. enabled = true
       mdi. mnufile. enabled = true
   End If
End Sub
```

通过以上菜单内容的学习,不难设计出如图 7-6 所示的菜单系统。该菜单系统运行情况如图 7-6 所示。

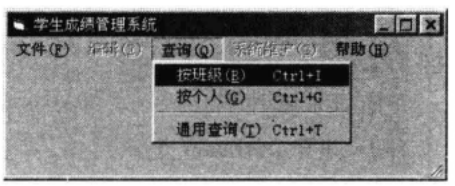


图 7-6 "学生成绩管理系统"菜单

7.3 弹出式菜单

大部分 Windows 应用程序都提供快捷菜单(弹出式菜单)以方便用户的使用,因为操作时只单击鼠标右键就可以调出快捷菜单,然后选择某一个菜单项,不需要再将鼠标移到窗口顶部去打开一层一层的菜单了。VB 中称快捷菜单为弹出式菜单,它实际上是独立于菜单栏而显示在窗体上的浮动菜单,通常将与鼠标所指对象密切相关的操作集中在快捷菜单中。

弹出式菜单的设计仍然还是采用菜单编辑器,与普通菜单不同的是,为了不将该菜单显示

在窗体上，需要将菜单标题设置为隐藏。不过，按照弹出式菜单的通常含义，鼠标所指对象不同，其快捷菜单的内容也是不同的。因此，就需要针对不同窗体的不同对象编制不同的弹出式菜单。

由于程序开始运行时，弹出式菜单是隐藏的，只有在指定对象上单击右键时才能弹出。因此，必须在程序代码中使用 PopupMenu 方法，才能将指定的快捷菜单弹出。PopupMenu 方法的基本语法格式如下：

```
object. PopupMenu menuname, flags, x, y
```

menuname 为弹出式菜单的名称(即菜单标题的名称)。参数 x，y 表示弹出式菜单的显示位置，默认为鼠标指针的当前位置。flags 进一步设置弹出式菜单的位置与行为，一般只要设置其行为。flags 取 0 时表示只能用鼠标左键来选择弹出式菜单中的项目，取 2 时则表示既可以用左键也可以用右键。

PopupMenu 方法一般是放在相关控件的 MouseDown 和 MouseUp 事件过程中。这两个事件是专门用于检测是否有鼠标按键按下(MouseDown)或释放(MouseUp)的。它们与 Click 事件有两点不同：一是这两个事件可以区分鼠标的左键或右键；二是它们也可以响应键盘 - 鼠标的组合操作，即按下键盘上特定的控制键(Ctrl、Shift 和 Alt 等)的同时再按动鼠标。这两个事件的语法格式是相同的，但比较复杂，其基本语法格式如下(以 MouseDown 为例)：

```
Private Sub object_MouseDown(button As Integer, shift As Integer)
```

button 返回一个整数，用来标识产生该事件是按下了哪个鼠标按键，1 表示左键，2 表示右键。shift 也是返回一个整数，在 button 参数指定的按键被按下的情况下，该整数将返回键盘控制键的状态。如果是 1 表示 Shift 键被按下，2 表示 Ctrl 键被按下，4 表示 Alt 键被按下。利用 shift 参数可以检测键盘 - 鼠标的组合操作。



不管是窗口顶部的菜单还是隐藏的菜单，都可以使用 PopupMenu 方法把它们作为弹出式菜单在程序运行期间显示出来。

【例 7-3】 弹出式菜单示例：为 frmExam_5_5 窗体中的“备选课程”列表框编写一个快捷菜单，包含两个菜单选项，即“添加选定”和“添加全部”，其作用与按钮“添加”和“全部添加”的功能类似。

操作步骤如下：

① 打开 frmExam_5_5 窗体，并启动“菜单编辑器”。

② 在“菜单编辑器”中输入所需的菜单项目，菜单标题“添加”，菜单名称为“pomTj”，并清除该项中“可见”检查框，如图 7-7 所示。

③ 编写菜单事件过程。由于本菜单在窗体上是不可见的，因此要对相关的菜单控件编写事件过程，只能先打开代码窗口，然后在对象列表框中选择菜单对象，再选择相关的事件过程。

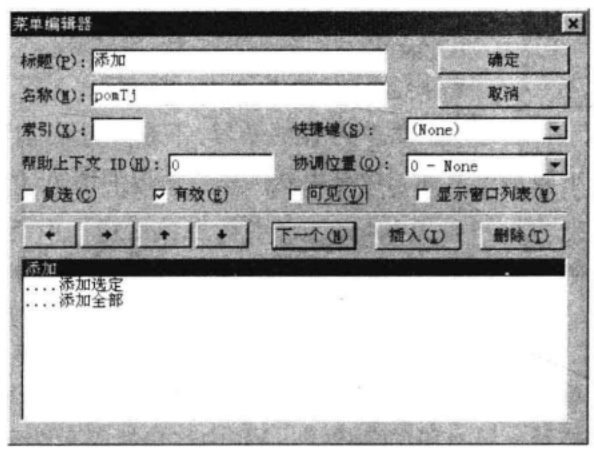


图 7-7　弹出式菜单的设计

• 当在“备选课程”列表框中右击鼠标时,要显示快捷菜单“pomTj”。显然,只要在 lstKc 列表框的 MouseUp 事件中调用 PopupMenu 方法即可。

• 当在弹出的快捷菜单中选择某一项时,需要调用相应按钮的功能。例如当选择了“添加选定”时,只要调用“添加”命令按钮的代码就行了,即,cmdTj. Value = True。

本例所涉及的相关代码如下所示:

```
Private Sub lstKc_MouseUp(Button As Integer, Shift As Integer, x As Single, y As Single)
  If Button = 2 Then                    '如果释放鼠标的右键则显示快捷菜单
    PopupMenu pomTj, 2
  End If
End Sub

Private Sub pomTj_Qb_Click()    '调用"全部添加"命令
    cmdQj. Value = True
End Sub

Private Sub pomTj_Xd_Click()    '调用"添加"命令
    cmdTj. Value = True
End Sub
```

④ 调试并运行程序。本例运行后的效果如图 7-8 所示。

思考 1: 如何为“拟选课程”列表框添加快捷菜单,也使其包含“清除选定”、“移除全部”菜单选项。

思考 2: 【例 7-2】在 MDI 窗体上设置了“课程设置”菜单的 Click 事件,请问为什么在单击“课程设置”菜单时调用 frmExam_5_5. frm(已在 lstKc 控件上完成了弹出式菜单的设计)窗体,如图 7-9 所示。

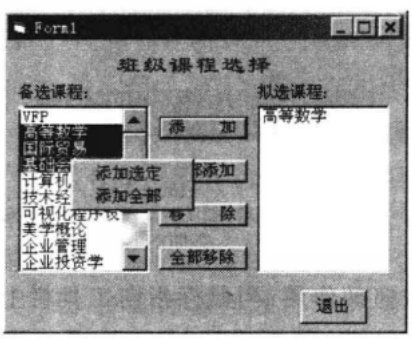


图 7-8 快捷菜单运行示例

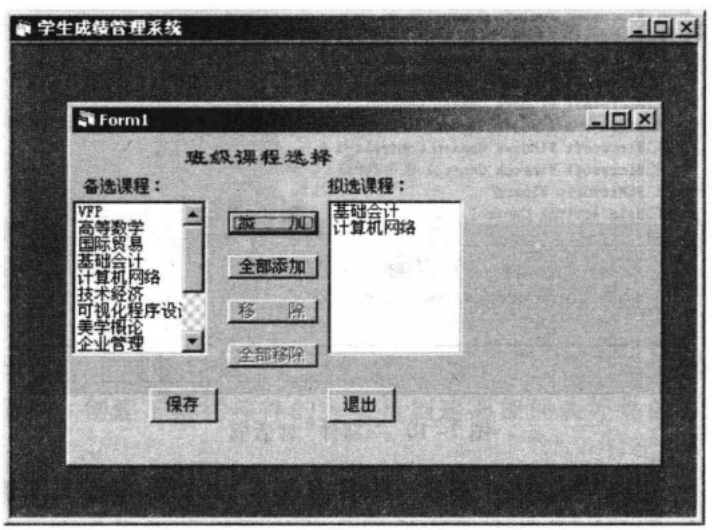


图 7-9 单击“课程设置”菜单后的界面

7.4　工　具　栏

【例7-4】 以本书的“学生成绩管理系统”为例，完成案例五中图7-1所示工具栏的制作，当按以左到右顺序单击工具栏上的按钮，对分别调出“注册”、“通用查询”、“选课”和“退出”窗体。

工具栏由多个按钮构成，它们通常放在窗体的菜单栏下。常用菜单的操作可以用单击工具栏的按钮实现，每个按钮一般对应菜单中的一项，从而免去查找菜单的时间，它提供了对应用程序中的快捷访问，极大地提高操作效率。

创建工具栏上的多个按钮与第2章案例二的简单记事本窗体上的按钮不同。VB是通过ToolBar和ImageList两个控件来创建工具栏，但是ToolBar和ImageList均不属于VB的内部控件，在标准控件工具箱中也就没有显示。因此，要使用这两个控件，必须先将其添加到控件工具箱中。方法是利用VB功能扩展方法（使用ActiveX控件方法）进行工具栏的设计。具体操作步骤如下：

ActiveX控件是VB工具箱的扩充部分，它是一个OCX文件，使用ActiveX控件的方法和使用工具箱中的20个标准控件方法一样，只是在使用时必须将它添加到工具箱中。

① 选择〖工程〗→〖部件〗命令或在控件工具箱面板上单击右键选择“部件”命令均会打开“部件”对话框，如图7-10所示。

② 选中“Microsoft Windows Common Control 6.0”，单击“确定”按钮。此时，在控件工具箱中会多出如图7-11所示的几个控件，其中包含了制作工具栏和状态栏所需要的两个控件。

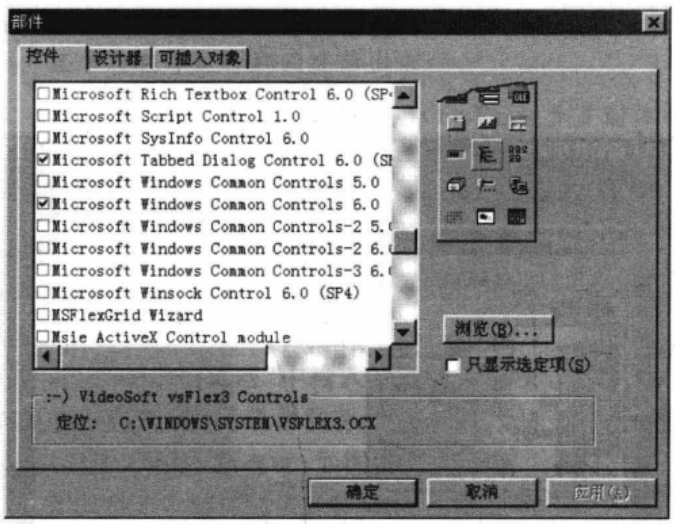


图7-10　“部件”对话框

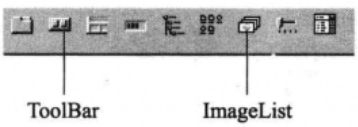


图7-11　添加的控件

工具栏上的每个按钮一般都有图片，从而形象地表明该按钮的作用，当鼠标指向工具栏上某个按钮时，还会出现该按钮的功能的提示说明文本。所有这些，都可以通过对ToolBar和ImageList这两个控件的有关属性进行设置来实现。

创建工具栏的一般步骤是：

① 创建 ImageList 控件，并在其中保存要使用的按钮图像。

② 创建 ToolBar 控件，并建立与 ImageList 对象的关联。

③ 在 ToolBar 控件中创建按钮对象。

④ 编写按钮对象的 Click 事件。

以下完成案例五中图 7-1 所示工具栏的制作，实现依次单击工具栏上的按钮分别调出“注册、通用查询、选课、退出”窗体的操作步骤。

1. 创建按钮图像

按钮上的图像一般都保存在 ImageList 控件中。ImageList 控件是一个特殊的控件，它专门为 Windows 公共控件保存图像。它提供了一份单一的图像目录，通过其 Key 属性或 Index 属性就可以引用其中的图像。

① 打开程序的 MDI 窗体，即包含菜单的 MDI 窗体。

② 双击控件工具箱中的 ImageList 控件，窗体上将出现该控件，将其名称改为 IMGTOOLS。

③ 右击 IMGTOOLS 控件，选择“属性”，再单击“图像”选项卡，显示其属性页。

④ 单击“插入图片”按钮，在“选定图片”对话框中选择所需要的图片（在 VB 的安装文件夹 Graphics\bitmaps\assorted 下的 key.bmp, card.bmp, notebook.bmp, bell.bmp），然后单击“打开”按钮（也可以选择多个图片同时插入）。

⑤ 对每个图片在其“关键字”框内输入一字符串作为该图片的 Key 属性值（也可不输入）。

⑥ 重复步骤④～⑤，直到将所需要的图片全部插入到该控件中，如图 7-12 所示。



图 7-12 插入图片后 IMGTOOLS 控件的属性页

在 VB 安装文件夹中有一个 Common 文件夹，在其中的 Graphics 文件夹中保存有大量各种各样的图片或图标文件，这些文件可以用在应用程序的开发中。

在 ImageList 控件中可以放置任意大小的图片或图标，但它们在其中的显示大小是相同的。通常，加入的第一幅图片决定了随后加入的图片大小。

2. 工具栏创建及其属性设置

只有在图像列表控件 IMGTOOLS 创建后，才可以创建工具栏控件，因为工具栏上的按钮要用到图像列表控件 IMGTOOLS 上的图片。创建工具栏控件的方法与创建图像列表控件的方法是相同的，操作步骤如下：

① 将鼠标指向控件工具箱中的 ToolBar 控件并双击，此时在窗体的上部菜单栏的下方将出现该控件，而且该控件会自动贯穿窗体的宽度。将所创建的控件名称改为所需要的名称，如 Toolbars。工具栏控件需要设置的属性较多，它们也是通过对其属性页进行设置

的。鼠标右击工具栏，选择“属性”即可显示“属性页”对话框。其中主要的设置如下：单击“通用”选项卡，在“图像列表”下拉框中选择 IMGTOOLS，以建立该工具栏与图像列表控件的关联，如图 7-13 所示。

② 单击图 7-13“按钮”选项卡。

③ 单击“插入按钮”以添加新的 Button 对象，此时在工具栏上将出现一个空白的按钮。

④ 在“图像”框中输入 IMGTOOLS 中相应的图像索引值，比如，1 即表示该按钮引用了图像列表控件中的第一幅图像。

⑤ 在“关键字”框中键入该按钮的 Key 属性值，通常可以与相应图像的关键字相同。这个值作为按钮的 Key 属性值，可以用来区别不同的按钮。

⑥ 在“工具提示文本”框中输入要提示的文本内容。这个内容用于介绍按钮的功能，以便在程序运行时，当用户的鼠标停留在按钮上时出现提示文本。

⑦ 单击“应用”按钮，在空白按钮上就会出现相应的图像。

⑧ 重复步骤③～⑦，直到将所有的按钮设置完毕，最后单击“确定”按钮退出。完成后如图 7-14 所示。

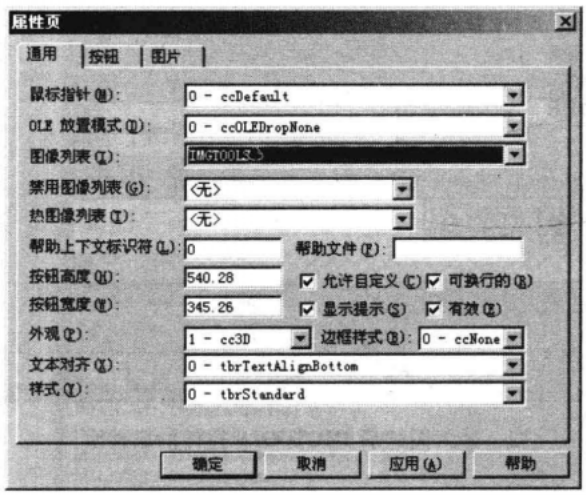


图 7-13　工具栏控件 Toolbars 的属性页(通用选项卡)

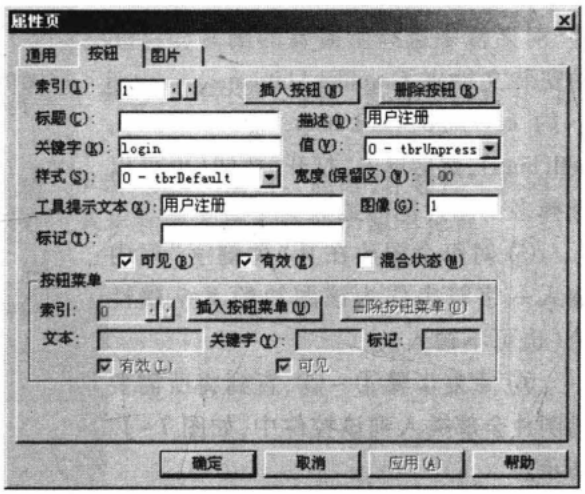


图 7-14　工具栏属性设置(按钮选项卡)

除了以上的主要属性设置以外，在属性页中还有一些其他的属性，例如，“样式”决定了按钮的风格(0 - 默认、1 - 检查、2 - 互斥按钮组、3 - 分隔符、4 - 位符)，“值”决定了按钮的初始状态(0 - 未按下、1 - 按下)，“按钮菜单”可以设置在单击按钮时弹出一个菜单。有关这些属性的设置方法，请参考有关的资料或 VB 的联机帮助。

3. 按钮事件编程

工具栏的按钮设置完成后，还需要进行相关的事件编程，否则程序运行时单击按钮是不会起作用的。当用户单击工具栏按钮时，会发生 ButtonClick 事件，该事件过程的语法格式如下：

```
Private Sub object_ButtonClick(ByVal button As Button)
```

其中的参数 Button 用来指明对按钮的引用，可以通过按钮的 Index 属性或 Key 属性来

判断用户单击了哪个按钮。通常情况下,使用 Key 属性比较可靠,因为用户可以用“自定义工具栏”对话框来重新安排 Button 对象,所以 Index 属性值可能并不一定能指向正确的按钮位置。

本例中的事件过程如下:

```
Private Sub Toolbars_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Key
    Case "login"
        MNUFILE_LOGIN_Click
    Case "find"
        MNUFIND_UNIVERSE_Click
    Case "selectbook"
        MNUALTER_KC_Click
    Case "exit"
        MNUFILE_EXIT_Click
End Select
End Sub
```



上述代码中的 Case "login"、Case "find"、Case "selectbook"、Case "exit" 的 key 值一定要与工具栏中相应按钮 Key 属性相同,即大小写是有区别的。

到此为止,案例五的工具栏制作才算完成。保存后运行,单击工具栏上的按钮即可调出相应的窗体(与案例五单击菜单的效果相同)。

习 题 7

1. 何谓 MDI 多文档界面? MDI 子窗体与普通窗体有何不同?
2. 在 MDI 窗体卸载时,如何及时保存子窗体中的信息?
3. QueryUnload 事件在什么时候被触发? 有何用途?
4. 试说明如何规划菜单项目?
5. 在 Windows 应用程序的菜单中,菜单项目有哪些表现形式? 在 VB 中如何实现?
6. 何谓菜单的访问键和快捷键? 如何设置?
7. 在菜单运行时可以对菜单进行哪些设置?
8. 菜单控件与普通控件有何异同? 如何编写菜单控件的事件过程?
9. 什么是弹出式菜单? 它与普通菜单有何不同? 如何显示弹出式菜单?
10. MouseDown 事件和 MouseUp 事件何时被引发? 它们与 Click 事件有何不同? 如何利用这两个事件来显示弹出式菜单?
11. 说明在 VB 中创建一个工具栏的一般步骤。
12. 如何将 ImageList 控件中的图像与工具栏控件中的按钮联系起来?
13. 在 ImageList 控件中可以放置大小不同的图像吗? 如何处理?
14. 在对按钮事件进行编程时,通常可以用 Key 属性来指明按钮而不用 Index 属性,为什么?

15. 将第2章案例二中的简易记事本中的命令按钮用本章所学的菜单及工具栏的知识实现相应功能。

16. 试设计一个窗体，其上有2个多行文本框和1个结束按钮，并为2个文本框设计弹出式菜单，以完成在2个文本框之间实现剪切、复制和粘贴功能。

17. 完成本章案例五的菜单及工具栏的制作。

18. 回答本章7.3节的思考1及思考2的问题。

第8章 文件管理

学习目标

掌握文件的基本概念、文件的结构与分类，熟悉读写顺序文件与随机文件的编程方法，了解文件相关的操作命令与函数。熟练掌握文件管理控件的使用方法，掌握使用通用对话框生成如打开对话框、另存为对话框、字体对话框的设置与编程。

任何一种应用程序一般总要对其相关的文件进行各种各样的处理，为了提高应用程序的开发效率，VB 提供了强大的对文件系统的支持能力。VB 提供了一整套的文件处理命令用于对文件进行各种处理。利用这些命令，用户可以很方便地访问文件系统并对文件进行读写操作。同时在程序中可以使用文件管理控件选择不同盘符不同目录下的文件进行相关操作，也可通过使用通用对话框控件来交互式地打开文件、保存文件。

【案例六】 文件操作。

本案例为演示性案例，为区别起见，将本案例的相关文件存放在一个新的工程文件 vb8_1. vbp 中，该工程有一个 MDI 窗体（见图 8-1），两个 MDI 子窗体（分别如图 8-2 和图 8-3 所示）。



图 8-1 MDI 窗体

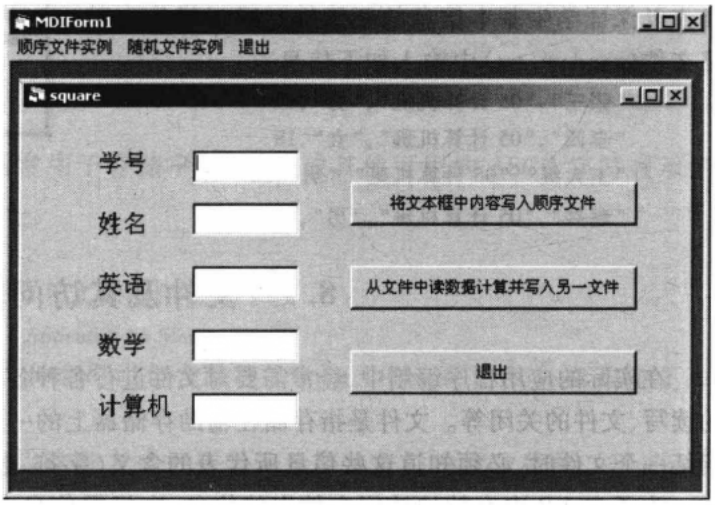


图 8-2 顺序文件实例窗体

在 MDI 窗体上建立如图 8-1 所示的菜单，通过单击相应的菜单分别调用两个子窗体（分别如图 8-2 和图 8-3 所示）。

在图 8-2 所示的子窗体单击第一个命令按钮可以将 5 个文本框中的内容以追加的方式写到顺序文件（score. txt）中；当单击第二个命令按钮可以将文件（score. txt）中的内容顺序读出并计算每个人的总分及平均成绩，然后顺序地写入文件（scorecount. txt）中。

单击第一个命令按钮，输入一组数据，然后再单击第二个命令按钮，打开两个文件如图 8-4

和图 8-5 所示。

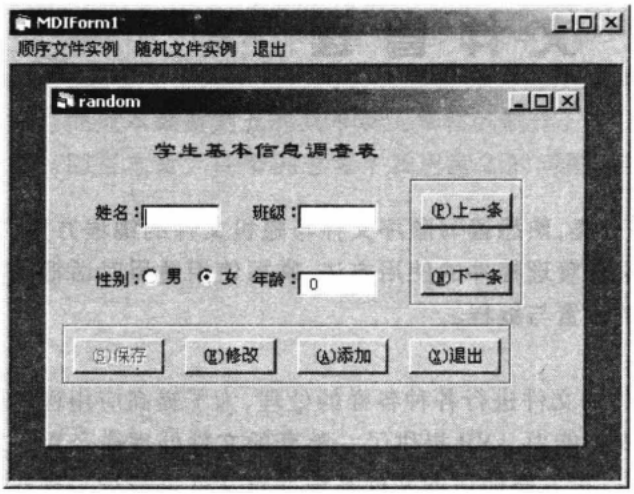


图 8-3　随机文件实例窗体

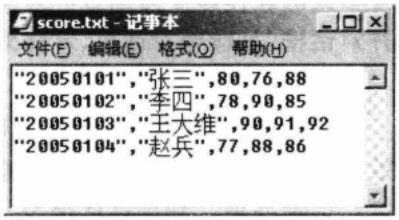


图 8-4　录入数据文件 score. txt

在图 8-3 所示的子窗体单击添加命令按钮可以在窗体上输入学生基本信息，单击保存按钮可以将窗体上的信息以追加方式写到随机文件中(random. txt)中；单击上一条或下一条可以浏览文件的记录内容，发现错误可以单击修改命令按钮进行修改以保证学生基本信息的正确性。通过操作在随机文件(random. txt)中输入如下信息：

"张三","05 计算机班","男",19
"李四","05 计算机班","女",18
"王大维","05 计算机班","男",20
"赵兵","05 计算机班","男",17

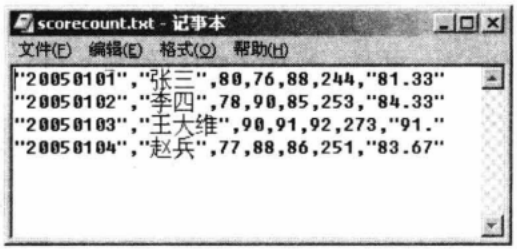


图 8-5　计算结果文件 scorecount. txt

8.1　文件及其访问方式

在实际的应用程序编制中，经常需要对文件进行各种各样的操作，如文件的创建或打开、文件的读写、文件的关闭等。文件是指存储在辅助存储器上的一系列相关信息的集合。当应用程序访问某一个文件时，必须知道这些信息所代表的含义(字符、数据记录、整数等)。

为了在 VB 中有效地访问文件中的信息，应根据信息在文件中的存放方式选择合适的文件访问类型。在 VB 中有 3 种文件访问的类型：顺序型、随机型和二进制型。

(1) 顺序型

它适用于访问普通的顺序文本文件。顺序文本文件也称为字符流式文件。该文件中的每一个字符都是由一个文本字符或文件格式字符(回车、换行等)所组成。

(2) 随机型

它适用于访问具有固定长度结构的文本文件或二进制文件，这类文件也称为随机文件。随机文件由一组相同长度的记录组成，记录可以由标准数据类型的单一字段组成，或者由多个字

段组成。每个字段的数据类型和长度可以不同,但文件每条记录的长度是相同的。数据一般是以二进制的形式存储在文件中。随机型的访问类型允许访问随机文件的任何记录。

(3) 二进制型

它适用于访问具有任意结构的文件,因此这也是一种最通用的访问类型。它实际上是以字节为单位对文件进行访问的,而不管是什么文件都可以认为是由字节构成的。它在访问时可以将文件指针移到任何位置。

本章将主要讨论前两种类型的访问方式。

8.2 文件的处理

为了适应应用程序对文件进行处理的要求,VB 为文件的处理提供了两套方案:一套是使用传统的语句与调用函数;另一套是使用 FSO(File System Object,文件系统对象)。FSO 以对象方式来处理文件和文件夹,利用 FSO 对象模型使得应用程序能够创建、改变、移动和删除文件夹,检测是否存在指定的文件夹。FSO 对象模型也能方便地获取关于文件夹的信息,诸如名称、创建日期或最近修改日期等。本书以第一套方案来介绍文件的处理,对于 FSO 有兴趣的读者可以参考相关的资料。

无论是顺序文件还是随机文件,文件的处理步骤基本都是相同的:

① 打开文件。

② 读写文件。

③ 关闭文件。

8.2.1 顺序文件的处理

顺序文件是普通的纯文本文件,通常用于存储字符、数字或其他可用的 ASCII 字符表示的数据,但不能存储像位图这样的信息。

1. 顺序文件的打开与关闭

对文件处理之前都要先打开文件,打开文件的语法如下:

```
Open pathname For [Input | Output | Append] As filenumber
```

其中,Pathname 为需要打开的文件名,文件名可以包含路径。Input | Output | Append 决定了文件的打开方式。

- Input:以输入的方式打开文件,以这种方式打开时表示要从文件中读取数据。因此,所指定的文件必须存在,否则会产生错误。
- Output:以输出的方式打开文件,表示要向打开的文件中写入数据。所写入的数据从文件头开始,文件原有的数据将被覆盖。如果所指定的文件不存在,将会创建一个新文件。
- Append:以追加的方式打开文件,表示要向打开的文件中追加数据。文件中的原有内容不会丢失,新的数据将追加到文件的最后。如果指定的文件不存在,则会自动创建一个新文件。

Filenumber 用于为打开的文件指定一个未使用的文件号。文件打开后对文件的一切处理都以文件号来表示该文件,直到关闭。文件号是一个介于 1 ~ 511 之间的整数,每个打开的文件对应一个唯一的文件号。为了保证打开的文件号不重复,VB 提供了一个获得下一个未被使用的

文件号的函数 FreeFile。例如：

```
Dim intFnum As Integer
intFnum = FreeFile                                    '获得未使用的下一个文件号
Open "D:\Progras\kcap.txt" For Input As intFnum       '以输入方式打开文件
```

文件的关闭非常简单，只要执行 Close 命令就行了。Close 命令的语法格式如下：

```
Close [filenumberlist]
```

其中，filenumberlist 为文件号列表，它们与在打开时指定的文件号是对应的，各个文件号之间用逗号隔开，如果省略该参数将关闭所有打开的文件。

2. 顺序文件的写入

当顺序文件以 Output 或 Append 方式打开后，就可以将数据写入到该文件中去。写入数据的方法有两种：Print #语句和 Write #语句。两个语句的区别是：Print #可以对写入的数据格式进行控制；而用 Write #写入的数据便于用 Input #语句来读取。

(1) Print #语句

Print #语句的语法格式如下：

```
Print #filenumber, [outputlist]
```

其中，filenumber 为需写入数据的文件号，outputlist 为需写入的数据列表，列表中的每个数据项用逗号或分号分开。如果数据项之间用逗号，表示用标准格式写入，即将每项写在一个打印区内(打印区的默认宽度为 14 个字符)；如果数据项之间用分号，表示用紧凑格式写入，即将所有数据项连续写入，不留空格。outputlist 如果省略，则在文件中写入一个空行。每个 Print # 语句所写入的数据在文件中占一行，将所有的数据项写完后自动换行。

outputlist 项省略后，其前面的逗号不能省略。

例如：

```
Open "D:\vbprogs\demo.txt" For Output As 1#          '以建立新文件的方法打开文件
Print #1,"数据写入演示:"                              '写入一个数据
Close                                                '关闭文件

Open "D:\vbprogs\demo.txt" For Append As 1#          '以追加方式打开文件
Print #1,"高等数学";"国际贸易"                        '数据项之间用分号分隔，紧凑格式
Print #1,"计算机网络","技术经济"                      '数据项之间用逗号分隔，标准格式
Print #1,                                            '写入一个空行
Print #1,txtKc.Text                                  '将文本框的内容写入文件
Close
```

(2) Write #语句

Write #语句的语法格式如下：

```
Write #filenumber, [outputlist]
```

从该命令的语法格式来看，它与 Print #语句的格式完全相同。它们之间的主要区别在写入数据的格式不同。

① 用 Write#语句写入的数据以紧凑格式存放，但在各个数据项之间会自动插入逗号作为分隔符。

② 如果写入的是字符串，Write #语句会自动在其首尾加上双引号作为数据的定界符。

用 Write #语句写入到文件中的数据，在读取时需要使用 Input #命令。例如，以下代码完成一批数据的写入：

```
Open "D:\Vbprogs\demo.txt" For Output As 1#          '以建立新文件的方法打开文件
Write #1,"数据写入演示:"                              '写入一个数据
Close                                                '关闭文件
Open "D:\Vbprogs\demo.txt" For Append As 1#          '以追加方式打开文件
Write #1,"高等数学:",75,"政治:",89
Write #1, "英语:",98,"计算机:",88
Write #1,                                            '写入一个空行
Write #1, "VFP:",86,"操作系统:",58
Close
```

此时在 D 盘的 Vbprogs 文件夹下将创建了一个新文件 demo.txt，其内容为：

```
"数据写入演示:"
"高等数学:",75,"政治:",89
"英语:",98,"计算机:",88

"VFP:",86,"操作系统:",58
```

【例 8-1】 设计案例六中各窗体及各窗体的控件属性，并编写图 8-2 中第一个命令按钮的 Click 事件代码，将图 8-2 中文本框中的内容以追加的方式写到顺序文件(score.txt)中。

各窗体及其控件的属性如表 8-1 所示。

表 8-1 工程 vb8_1 中窗体及控件属性

窗体	控件名	属性名	设置值
MDI	MDI 窗体	Name	MDI8_1
	菜单 1	Caption	顺序文件实例
		Name	square
	菜单 2	Caption	随机文件实例
		Name	random
	菜单 3	Caption	退出
		Name	exit
顺序文件实例	顺序文件实例子窗体	Name	frmexam_8_1
		Caption	square
	标签 1	Caption	学号
	标签 2	Caption	姓名
	标签 3	Caption	英语
	标签 4	Caption	数学
	标签 5	Caption	计算机
	文本框 1	Name	Text1(0)
	文本框 1	Name	Text1(1)

续表

窗体	控件名	属性名	设置值
顺序文件实例	文本框1	名称	Text1(2)
	文本框1	名称	Text1(3)
	文本框1	名称	Text1(4)
	命令按钮1	名称	cmdok
		Caption	将文本框中内容写入顺序文件
	命令按钮2	名称	read_count_write
		Caption	从文件中读数据计算并写入另一文件
	命令按钮3	名称	cmdexit
		Caption	退出

在案例六中图8-2中第一个命令按钮的Click事件代码如下：

```
Private Sub cmdok_Click()
    no = Text1(0)
    sname = Text1(1)
    english = Val(Text1(2))
    math = Val(Text1(3))
    computer = Val(Text1(4))
    If no = "" And sname = "" And english = 0 And math = 0 And computer = 0 Then
    Exit Sub
    Else
    Open "D:\Vbprogs\score.txt" For Append As #10      '以追加方式打开顺序文件
    Write #10, no, sname, english, math, computer      '将各变量值写到打开的文件中
    Close                                              '关闭打开的文件
    For i = 0 To 4                                     '以下代码是清除文本框的内容
            Text1(i).Text = ""
    Next i
    Text1(0).SetFocus
  End If
 End Sub
```

3. 顺序文件的读取

要读取顺序文件的内容，必须先以Input方式打开。顺序文件打开后，就可以读取文件的内容了。VB提供了3种读取顺序文件内容的方法：

- Line Input #filenumber, varname
- Input #filenumber, varlist
- Input(number, filenumber)

在这3种读取方法中，filenumber均为文件号。

第一种方法每次从指定的文件中读取一行(从当前字符开始到回车符或换行符之前的所有字符)数据，并将读取的数据送到字符串变量varname中。

第二种方法可以将文件中的数据读到 varlist 指定的多个变量中，变量表中变量的个数和类型必须保证与文件中读出数据的个数和类型相匹配，否则会出错。

第三种方法是函数调用的方法，也是使用最多的文件读取手段之一，它可以从文件中一次读取指定长度(number)的文本，而且它还能读取包括换行符、回车符和空格等各种字符。

在对文件进行读取时常常要判断文件的长度与文件结束符，为此可以使用 VB 所提供的两个函数 EOF 与 FileLen。

(1) EOF 函数

在读取文件内容时，通常使用文件指针来指明读取的位置。该函数用于测试文件指针是否已经位于文件的尾部，如指向文件的尾部则返回真，否则返回假。语法格式为：

```
EOF(filenumber)
```

(2) FileLen 函数

该函数是以长整数的形式返回某个文件的长度(字节数)，语法格式为：

```
FileLen(pathname)
```

Pathname 为文件名，可以包括路径。要注意的是，如果指定的是一个已经打开的文件，其返回的是该文件打开前的长度。与该函数类似的还有一个 LOF 函数，它也返回文件的长度，但它与 FileLen 函数不同的是它必须在文件用 Open 语句打开后才能使用。

【例 8-2】 顺序文件的读取。在 D 盘的 Vbprogs 目录下有一个 kcap. txt 的文件，存放了课程安排的内容，每行一种课程，试将该文件的内容读出并放置到一个列表框中。程序代码如下：

```
Dim intFnum As Integer
Dim strKc As String
intFnum = FreeFile
Open "D:\Vbprogs\kcap. txt" For Input As intFnum    '以 Input 方式打开文件
Do Until EOF(1)                                    '判断是否已经读到文件尾部
    Line Input intFnum, strKc                      '每次读取一行
    lstBjkc. AddItem strKc + Chr(13) + Chr(10)'将读取的内容添加到列表框中,并加上回车符与换行符
Loop
Close intFnum
```

【例 8-3】 顺序文件的读写。

完善例 5-5(对应的文件为 frmExam_5_5)，要求在窗体启动时首先检查在 D 盘的 Vbprogs 文件夹下是否存在 kcap. txt 文件。如果存在则读出文件的内容并显示在“拟选课程”列表框中；如果不存在则创建该文件，同时清除“选定课程”列表框。

在窗体上增加一个“保存”按钮，其作用是将“选定课程”列表框中的内容保存到 kcap. txt 文件中去。

分析：本题主要需要解决 3 个问题。

① 判断指定的文件是否存在？这可以通过 Dir 函数来实现。代码可放置在窗体的 Load 事件中。

② 如何读取文件的内容并显示？文件内容的读取用 Line Input 语句配合循环，每读取一行在“拟选课程”列表框中显示一行。读取完成后将文件关闭，然后再使用 Output 方式将该文件打

开,以备保存新的课程内容。该段代码也可以放置在窗体的 Load 事件过程中。

③ 如何保存"拟选课程"列表框的内容?列表框内容的保存同样要使用循环来实现:首判断列表框中是否有内容,如有的话将每个项目用 Print #语句保存到指定的文件中去。本代显然应放置在"保存"按钮的单击事件中。

新增的具体代码如下:

```
Private Sub Form_Load( )
    Dim intFnum As Integer
    Dim strKc, strFile As String
    strFile = Dir("D:\vbprogs\kcap.txt")          '判断指定的文件是否存在
    If strFile < > "" Then                        '如果存在,则打开读取内容并显示在列表框中
            Open "D:\Vbprogs\kcap.txt" For Input As #1
            Do Until EOF(1)
            Line Input #1, strKc                  '读取内容
            lstBjkc.AddItem strKc                 '显示内容
            Loop
            Close #1
    Else
    MsgBox ("所指定的文件不存在,系统将新建该文件", vbOKOnly, "警告")
    End If
    Open "D:\Vbprogs\kcap.txt" For Output As #1   '以 Output 方式重新打开
End Sub
Private Sub cmdSave_Click( )                      '"保存"按钮的单击事件过程
    If lstBjkc.ListCount > 0 Then                 '如列表框中有项目,则将其内容保存到文件
        For i = 0 To lstBjkc.ListCount - 1        '循环保存
            Print #1, lstBjkc.List(i)
        Next i
    End If
End Sub
```

【例 8-4】 编写在案例六图 8-2 中第二个命令按钮的 Click 事件代码。当单击第二个令按钮可以将文件(score.txt)中的内容顺序读出并计算每个人的总分及平均成绩,然后顺序写入文件(scorecount.txt)中。

案例六图 8-2 中第二个命令按钮的 Click 事件代码如下:

```
Private Sub read_count_write_Click( )
    Open "D:\Vbprogs\score.txt" For Input As #10
    Open "D:\Vbprogs\scorecount.txt" For Output As #11
    Do While Not EOF(10)      '循环将 10 号文件的每行信息读出并计算然后写到 11 号文件中
    Input #10, no, sname, english, math, computer
    sum = english + math + computer
    pj = Format(sum / 3, "##.##")
    Write #11, no, sname, english, math, computer, sum, pj
Loop
Close
```

```
End Sub
```

8.2.2 随机文件

使用顺序文件的方式对数据进行存取是比较常用的,有时也是比较方便的,但顺序文件有一个致命的缺陷,那就是对顺序文件只能采用顺序方式进行存取。如果文件的数据量很大,而且还要经常地进行数据的查找与修改,这时用顺序文件来实现是非常困难的。事实上,在应用程序的开发中经常要对各种信息进行管理,如:成绩信息、工资信息和档案信息等。使用随机文件对这类信息进行管理是非常方便而有效的。

随机文件是以记录为单位进行存取的,而且每个记录的长度是固定的,每条记录都有一个记录号,记录又由一个或多个字段(也称为域)构成,如:学生信息可以由姓名、班级、性别、年龄等域所构成。随机文件打开后,可以根据记录号访问任何一条记录,既可读又可写。

1. 记录类型的定义

通常随机文件的记录都是由多个字段构成的,而且各个字段的数据类型与宽度也可各不相同。因此,在使用随机文件时必须对记录的数据结构进行定义。

(1) 定义记录类型

定义记录类型是用 Type 语句来实现的,其语法如下:

```
[Private | Public] Type varname
    elementname   As type
    [elementname   As type]
    ...
End Type
```

elementname 为记录中字段的名称,type 为字段的数据类型。varname 为记录类型的名称。

例如,以下定义了一个有关学生情况信息的记录类型:

```
Public Type Stu_info
    strXm As String * 8   '学生姓名
    strBj As String * 20  '学生班级
    strXb As String * 2   '学生性别
    intNl As Integer      '学生年龄
    strTc As String * 30  '学生特长
End Type
```

在进行记录类型的定义时要注意以下问题:

① 只能在标准模块的声明部分进行定义,默认为 Public。

② 如果其中的某些成员为字符串,则必须是定长字符串。

③ 记录类型的长度取决于其每个成员的长度,通常等于所有成员长度之和。

④ 类型的名称与变量的名称不同,前者与 Integer、String 等是类似的;而对于后者 VB 必须根据其变量的类型来分配所需的空间以存储数据。

类型定义完成后,即可定义使用该类型的变量了。

(2) 记录变量的定义

记录变量可以在相应的程序段的声明中进行定义,它与一般的变量定义是相同的,如:

```
Dim Stu As Stu_Info
```

该语句定义了一个 Stu_Info 类型的 Stu 变量,利用该变量就可以来存取相应的随机文件了。

2. 随机文件的打开与关闭

随机文件的打开与关闭仍然使用 Open 和 Close 语句。要打开随机文件必须使用以下格式的 Open 语句:

```
Open pathname [For Random] As filenumber [Len = reclength]
```

For Random 指定用随机方式打开文件,这也是一种缺省的访问方式,因此可以省略。Len = reclength 指明所打开的随机文件的记录长度,该长度必须等于记录类型定义时的长度,同时一般也应等于文件中记录的实际长度。

记录长度的确定可以使用 Len 函数,其用法为:Len(varname)。其中 varname 为记录变量名。

例如:

```
Dim Stu As Stu_Info                                          '定义记录变量
Dim intFnum, intLast As Integer                              '文件号与最大记录数变量
intFnum = FreeFile                                           '得到未使用的文件号
Open "D:\Vbprogs\xsxx.dat" For Random As intFnum Len = Len(Stu)  '以随机方式打开文件,记录长度由
                                                             'Len 函数得出
```

随机文件一经打开就可读写。至于关闭随机文件的语句 Close 的用法与前述完全相同,不再赘述。

3. 随机文件的读写

随机文件的读写分别使用 Get #语句和 Put #语句进行。

(1)写入记录

向随机文件中写入记录使用 Put #语句,其语法格式为:

```
Put [#]filenumber, [recnumber], varname
```

其中,varname 为要写入的记录变量名。recnumber 为记录号,表示要将数据写入到第几个记录中。记录号的有效范围为 $1 \sim 2^{31}-1$,随机文件中的第一个记录其记录号始终为 1,第二个记录的记录号为 2,依次类推。记录号参数是可以省略的,如果省略则从上一个 Get 或 Put 语句确定的下一个记录或上一个 Seek 函数所指出的记录开始写入,记录号省略后,其前后的逗号是不能省略的。

Put #语句主要用于对随机文件的记录进行添加和修改。

例如:将 xsxx.dat 文件用上述随机方式打开后,用如下的命令即可将记录变量的值写入到文件中的第 1 号记录位置上。

```
With Stu
    .strXm = "王大维"
    .strBj = "01 级计算机"
    .strXb = "男"
    .intN1 = 23
    .strTc = "体育、乐器"
End With
Put #intFnum,1,Stu                '写入 1 号记录
```

如果要将新记录写在文件的尾部,可以先求出文件中现有的记录数,记录数加 1 即为要写入新记录的记录号。将上述代码的最后一句改为如下代码即可实现。

```
intLast = Lof(intFnum)/Len(Stu)          '文件的总长度除以记录长度即为记录数
```

```
intLast = intLast + 1
Put #intFnum, intLast, Stu
```

如果所指定的记录号在文件中已经存在,则在写入时将替换原有的记录,而其他记录内容不变。

(2) 记录的定位

在随机存取方式中,可以存取随机文件中的任意一条记录,前提是要进行准确的记录定位。利用 Seek 函数和命令可以方便实现记录的定位。

- Seek 函数:该函数的功能是返回当前记录号,其语法格式为:

```
Seek(filenumber)
```

返回值也在 $1 \sim 2^{31} - 1$ 之间。

- Seek 命令:该命令的功能是设置新的当前记录以供读写。其语法格式为:

```
Seek [#]filenumber, position
```

position 指出下一次要读写的记录号,但应注意,其值一般不要超过文件所包含记录的有效范围,其值为 $1 \sim 2^{31} - 1$。

当使用 Seek 命令来进行读写记录的定位时,在 Put 或 Get 语句中记录号参数必须省略,否则将以读写命令中的记录号为准。

(3) 记录的读取

使用 Get #语句可以将记录内容复制到记录变量中。该语句的语法是:

```
Get [#]filenumber, [recnumber], varname
```

从语法可以看出,Get #语句与 Put #语句无论是格式还是用法都非常类似。所不同的是前者是从文件中读取记录,而后者是向文件中写入记录。

使用 Get 语句将记录内容读出后,就可以对记录进行进一步的处理,如将记录的内容显示在相应的文本框中。

【例 8-5】 随机文件的处理。完成在案例六如图 8-3 子窗体中单击“添加”按钮可以在窗体上输入学生基本信息(包括姓名、班级、性别和年龄),单击保存按钮可以将窗体上的信息以追加方式写到随机文件(random. txt)中;单击“上一条”或“下一条”可以浏览文件的记录内容,发现错误可以单击修改命令按钮进行修改以保证学生基本信息的正确性。

图 8-3 子窗体及其控件的属性如表 8-2 所示。

表 8-2 图 8-3 子窗体及控件属性

控 件 名	属性名称	设 置 值
随机文件实例子窗体	Name	frmExam_8_2
	Caption	random
文本框 1	Name	txtXm
文本框 2	Name	txtBj
文本框 3	Name	txtN1
单选按钮 1	Name	OptXb(0)
	Caption	男

续表

控件名	属性名称	设置值
单选按钮2	Name	OptXb(1)
	Caption	女
命令按钮1	Name	cmdBrow(0)
	Caption	(&P)上一条
命令按钮2	Name	cmdBrow(1)
	Caption	(&N)下一条
命令按钮3	Name	cmdGrp1(0)
	Caption	(&X)退出
命令按钮4	Name	cmdGrp1(1)
	Caption	(&A)添加
命令按钮5	Name	cmdGrp1(2)
	Caption	(&E)修改
命令按钮6	Name	cmdGrp1(3)
	Caption	(&S)保存

案例六图8-3子窗体中各命令按钮的程序代码如下：

```
'以下是定义记录类型,该代码存放在标准模块 Stu_information 中
    Public Type Stu_info
      strXm As String * 8
      strBj As String * 20
      strXb As String * 2
      intN1 As Integer
    End Type
```

```
'以下是案例六图 8-3 子窗体中各命令按钮的程序代码
    Option Explicit
    Dim Stu As Stu_info                          '定义记录变量
    Dim Reclen, intFnum, intLast, intRec As Integer
```

```
    Private Sub cmdBrow_Click(Index As Integer)      '"上一条"、"下一条"按钮事件过程
        Select Case Index
        Case 0                                       '上一条记录
        If intRec > 1 Then
            Seek #intFnum, intRec - 1                '设置当前读写记录号
        End If
      Case 1                                         '下一条记录
        If intRec < LOF(intFnum) / Len(Stu) Then     '测试当前记录是否超过最大记录号
            Seek #intFnum, intRec + 1
        End If
```

```
    End Select
    intRec = Seek(intFnum)                    '返回当前记录号
    Get #intFnum, , Stu                       '将当前记录读入变量 Stu
    Seek #intFnum, intRec                     '设置当前记录号
    dispXsxx                                  '调用记录信息显示过程
End Sub
```

```
Private Sub cmdGrp1_Click(Index As Integer)   '命令组按钮事件过程
    Dim strXbz As String
    Dim i As Integer
    Select Case Index
        Case 0
            Close #intFnum                    '关闭文件
            Unload Me
        Case 1                                '添加记录
            cmdGrp1(3).Enabled = True         '以下为设置按钮的可操作与否
            cmdGrp1(2).Enabled = False
            cmdGrp1(1).Enabled = False
            cmdBrow(0).Enabled = False
            cmdBrow(1).Enabled = False
            intLast = LOF(intFnum) / Len(Stu) '文件的总长度除以记录长度即为记录数
            intRec = intLast + 1
            txtXm.Text = ""                   '清空控件内容
            txtBj.Text = ""
            txtN1.Text = ""
        Case 2                                '修改记录
            cmdGrp1(3).Enabled = True         '以下为设置按钮的可操作与否
            cmdGrp1(1).Enabled = False
            cmdGrp1(2).Enabled = False
            cmdBrow(0).Enabled = False
            cmdBrow(1).Enabled = False
            intRec = Seek(intFnum)
        Case 3                                '保存记录
            If optXb(0).Value = True Then     '判断性别的值
                strXbz = optXb(0).Caption
            Else
                strXbz = optXb(1).Caption
            End If

            With Stu                          '将控件的值赋予记录成员
                .intN1 = Val(txtN1.Text)
                .strBj = txtBj.Text
```

```
            .strXb = strXbz
            .strXm = txtXm.Text
        End With
        Put #intFnum, intRec, Stu                    '写入文件
        cmdGrp1(3).Enabled = False
        cmdGrp1(1).Enabled = True
        cmdGrp1(2).Enabled = True
        cmdBrow(0).Enabled = True
        cmdBrow(1).Enabled = True
    End Select
End Sub
```

```
Private Sub Form_Load()                              '窗体装载代码
    Reclen = Len(Stu)                                '返回记录长度
    intFnum = FreeFile                               '得到文件号
    '以随机方式打开文件
    Open "D:\vbprogs\random.txt" For Random As #intFnum Len = Reclen
    Get #intFnum, , Stu                              '读入第一条记录
    dispXsxx                                '调用信息显示过程,在窗体上显示第一条记录信息
    Seek #intFnum, 1                                 '重新定位记录
    intRec = Seek(intFnum)                           '保存当前记录号
End Sub
```

```
Private Sub txtN1_Validate(Cancel As Boolean)        '年龄值的有效性判断
    If Val(txtN1.Text) <= 0 Then
        MsgBox ("年龄应该由数字构成且不能小于零,请重新输入")
        Cancel = True                                '年龄值无效取消
    End If
End Sub
```

```
Public Sub dispXsxx()                                '记录信息的显示过程
    txtXm.Text = Stu.strXm
    txtBj.Text = Stu.strBj
    txtN1.Text = Str(Stu.intN1)
    If Stu.strXb = "男" Then
        optXb(0).Value = True
    Else
        optXb(1).Value = True
    End If
End Sub
```

8.2.3 文件处理常用命令及函数汇总

VB所提供的用于文件处理的命令与函数较多,为了便于学习,现将有关的命令、函数整理

在表8-3中。表中大多数命令、函数已经在前面进行了介绍,还有一些没有介绍的请读者参考有关资料或联机帮助。

表8-3 用于文件处理的命令与函数

分类	命令或函数主要格式	功 能	备 注
目录操作	CurDir[(drive)]	返回当前目录	
	ChDrive drive	改变当前驱动器	
	ChDir path	改变当前目录	
	MkDir\|RmDir path	建立或删除目录	
文件操作	Dir[(pathname)]	显示文件列表	
	FileCopy source, destination	文件复制	
	Kill pathname	文件删除	
	Open pathname For[Input \| Output \| Append] As filenumber	文件打开	用于顺序文件
	Open pathname [For Random] As filenumber[Len = reclength]	文件打开	用于随机文件
	Close [filenumberlist]	文件关闭	
	Line Input #filenumber, varname	读文件	用于顺序文件
	Input #filenumber, varlist	读文件	用于顺序文件
	Input(number, [#]filenumber)	读文件	用于顺序文件
	Get [#]filenumber, [recnumber], varname	读文件	用于随机文件
	Put [#]filenumber, [recnumber], varname	写文件	用于随机文件
	Write #filenumber, [outputlist]	写文件	用于顺序文件
	Print #filenumber, [outputlist]	写文件	用于顺序文件
	FileLen(pathname)	返回文件长度	文件未打开时
	FreeFile[(rangenumber)]	返回可用文件号	
	LOF(filenumber)	文件长度	文件打开后
	EOF(filenumber)	测试文件尾部	
	Seek [#]filenumber, position	设置读写位置	
	Seek(filenumber)	返回当前读写位置	
	Loc(filenumber)	返回最近一次的访问位置	

8.3 文件管理控件

在应用程序中,经常要对文件进行各种各样的操作(如选择不同磁盘不同目录下的文

件以确定文件的位置，然后进行文件管理操作，如复制、删除和改名等）。实现这样的操作首先要了解有关驱动器、目录及文件的相关信息。为此，VB 提供了驱动器列表框（DriveListBox）、目录列表框（DirListBox）和文件列表框（FileListBox）3 个控件用于实现这样的功能。

在 Windows 中，目录称为文件夹。考虑到一般通俗的说法，本章仍然将文件夹称为目录。

【案例七】 新建一个工程文件 vb8_2. vbp，该工程有一个窗体，在该窗体上使用文件系统的 3 个控件对计算机系统的文件进行复制操作。运行后的窗体如图 8-6 所示。

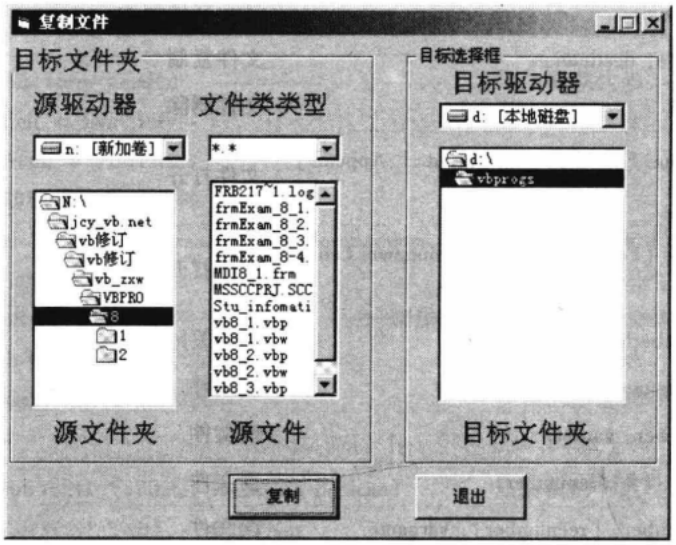


图 8-6　文件系统控件组合应用实现文件复制操作

8.3.1　驱动器列表框

要查找定位一个文件，必须知道 3 个要素：驱动器、路径和文件名，与其对应的 3 个文件管理控件可以满足文件查找定位的需要，如图 8-7 所示。

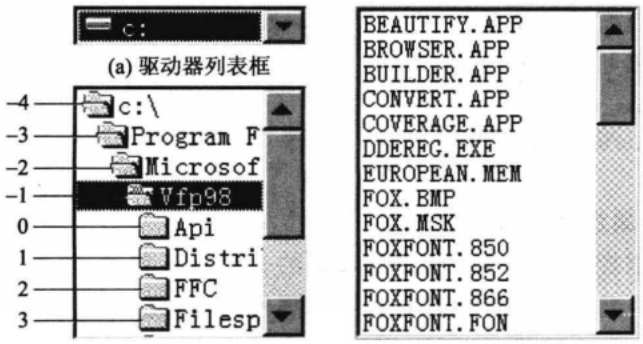
驱动器列表框是一个下拉式列表框。运行时，在默认情况下只显示当前驱动器的名称，单击下拉箭头会显示本地计算机上所有的驱动器，从而可以选择所需要的驱动器，用户也可以直接输入有效的驱动器字母来实现选择。新的驱动器一旦选定，该驱动器名称将出现在列表框的顶端，如图 8-7a 所示。

驱动器列表框控件的使用很简单，其属性在设计时除了名称以外无需进行任何设置。如果要返回用户所选择的驱动器名称可以使用 Drive 属性，该属性只能在运行时指定或返回。

例如，在窗体的 Load 事件中输入“drvVbxx. Drive = "D:"”可以将列表框中的默认驱动器改为 D 盘。

由于一般驱动器列表框控件都不是单独使用，它常常与目录列表框及文件列表框配合使

用,因此,在运行时必须根据用户的选择对目录列表框与文件列表框进行更新以保持一致。每当用户选择或指定了一个新的驱动器后,就会引发 Change 事件,因此可以在该事件中通过返回的 Drive 属性来更新其他两个控件。



(a) 驱动器列表框

(b) 目录列表框 (c) 文件列表框

图 8-7 文件管理控件运行示例

驱动器列表框中对驱动器的改变并不能改变系统的当前工作驱动器,如果想改变系统的当前驱动器,必须使用 ChDrive 命令。

8.3.2 目录列表框

目录列表框主要是用于显示当前或指定驱动器上的目录结构,它是一个列表框。显示时从根目录开始,各子目录按层次结构依次缩进,类似于 Windows 中的"打开"对话框。利用单击或光标键的移动可以选择某个目录,双击目录会打开该目录下的所有子目录,如图 8-7b 所示。

目录列表框的使用比较简单,设计时除了要设置名称外,无需进行其他的设置。关键是在运行时对其相关的属性进行必要的设置或返回有关的属性值。

1. 设置与返回当前路径

目录列表框的 Path 属性可以用来改变或设置当前路径,这个属性不能在设计时指定,而只能在运行中使用。其语法格式如下:

```
object. Path [ = pathname]
```

其中,pathname 是一个指定的路径名字符串。例如,dirVbxx. Path = "d:\Vbprogs"命令将当前路径改为 D 盘下的 Vbprogs 子目录。

2. 保持与驱动器列表框的同步

当驱动器列表框中的当前驱动器改变后,目录列表框也必须改为指定驱动器的当前目录以实现两个控件的同步。只要在驱动器列表框的 Change 事件中,将当前驱动器的名称赋予目录列表框的 Path 属性即可。代码如下:

```
Private Sub drvVbxx_Change( )
    dirVbxx. Path = drvVbxx. Drive
End Sub
```

目录列表框中对路径的改变并不能改变系统的当前目录,如想改变系统的当前目录必须使用 ChDir 命令。

3. 改变操作系统级的当前目录

当目录列表框的当前目录改变后,将引发 Change 事件,在该事件中就可以改变操作系统级的当前目录了。其代码如下:

```
Private Sub dirVbxx_Change( )
    ChDir dirVbxx. Path
End Sub
```

4. 利用 ListIndex 属性来改变当前目录

在第5章的列表框控件中曾经介绍过列表框的 ListIndex 属性,该属性常用来返回列表框中当前选择项目的索引。目录列表框也具有这样的属性,不过其含义稍有不同。在目录列表框中 ListIndex 主要用来标识每个目录。由 Path 属性所指定的目录的 ListIndex 的值始终是 -1,其上一级目录为 -2,再上一级为 -3……当前目录的第一级子目录的 ListIndex 值为0,向下依次为1、2、3等,如图 8-7b 所示。

利用 ListIndex 属性可以轻而易举地实现在目录中的上下移动。以下的代码即可以判断当前目录的上一级目录是否存在,如果存在,则将上一级目录设为当前目录。

```
If dirVbxx. List( -2) < > "" Then
    dirVbxx. Path = dirVbxx. List( -2)
End If
```

如果想了解目录列表框当前显示的目录下有多少直接的子目录,可以访问目录列表框的 ListCount 属性。

8.3.3 文件列表框

驱动器列表框与目录列表框可以定位文件所在的盘符与路径,但要显示所需要的文件就得借助于文件列表框了。文件列表框(FileListBox)也是一种列表框,用于显示 Path 属性指定目录中的所有文件或指定文件的清单,如图 8-7c 所示。

1. 设置或返回所显示文件的路径

显然,在文件列表框中只能显示某个路径下的文件。通过文件列表框的 Path 属性可以改变需显示的路径,当然也可以返回其路径。Path 属性只能在相应的事件代码过程中(如目录列表框或驱动器列表框的 Change 事件)使用,不能在设计时设置。例如,下面的代码将文件列表框 filVbxx 的 Path 属性改变到目录列表框 dirVbxx 中 Path 属性所指定的目录中。

```
Private Sub dirVbxx_Change( )
    filVbxx. Path = dirVbxx. Path
    ChDir dirVbxx. Path                    '将操作系统的当前目录改变到 dirVbxx 所指定的目录
    filVbxx. Pattern = " * . bas; * . frm"  '在文件列表框中只显示所有的 . bas 文件和 . frm 文件
End Sub
```

2. 显示指定模式的文件

默认时,文件列表框将显示 Path 属性指定目录中的所有文件,即其值为“ * . * ”。如果需

要显示指定模式的文件可以改变文件列表框的 Pattern 属性的值，该属性支持通配符“*”和“?”。例如：" *. txt" 显示所有的 . txt 文件，" *. bas; *. frm" 显示所有的 . bas 文件和 . frm 文件。Pattern 属性既可以在设计时设置，也可以在运行时设置。

3. 返回运行时所选择的文件名

文件列表框在运行时，用户单击其中某个文件，该文件会突出显示。如要返回用户所选择的文件名，使用文件列表框的 FileName 属性即可。显然，该属性在设计时也是不可用的。

由于文件列表框也是一种列表框，因此要返回所选择的文件名（即列表框的项目名），也可以使用 List(ListIndex) 属性。

FileName 属性不仅能返回所选择的文件名，还能设置文件列表框中需要显示的文件。在设置时，可以带文件的路径或盘符。例如，filVbxx. FileName = " D: \Vbprogs\ *. frm" 指定在 filVbxx 文件列表框中显示 D 盘 Vbprogs 下所有的 . frm 文件。

使用 FileName 属性并不能直接返回文件所在的路径，文件的路径必须通过 Path 属性单独检索。

当改变了文件列表框的 Path 属性或设置的 FileName 属性后，会触发文件列表框的 PathChange 事件；当改变了文件列表的文件显示模式后会引发 PatternChange 事件。

【例 8-6】 文件系统控件的组合使用完成案例七的文件复制任务。要求：使用文件系统的 3 个控件对计算机系统的文件进行浏览，选择源文件并选择希望将源文件复制到的目标位置后，单击“复制”按钮将选定的文件复制到目标盘下的目标目录中。

窗体及主要控件属性的设置如表 8-4 所示。

表 8-4 窗体及主要控件属性

控 件 名	属 性 名	设 置 值
“复制文件”窗体	Name	frmexam_8_3
	Caption	复制文件
“源驱动器”驱动器列表框	Name	Drive1
“源文件夹”列表框	Name	Dir1
“源文件”列表框	Name	File1
“文件类型”组合框	Name	Combo1
“目标驱动器”驱动器列表框	Name	Drive2
“目标文件夹”列表框	Name	Dir2
“复制”命令按钮	Name	Copy
	Caption	复制
“退出”命令按钮	Name	Exit
	Caption	退出

实现案例七的文件复制任务的代码如下所示：

```
Option Explicit
```

```
Dim strSorceFile, strDestinFile As String
Private Sub Combo1_Change()        '在下拉列表框选择文件类型
    File1. Pattern = Combo1. Text
End Sub
Private Sub Combo1_Click()         '接受输入的文件类型
   File1. Pattern = Combo1. Text
End Sub
Private Sub Command2_Click()       '退出按钮的 Click 事件
      Unload Me
End Sub
Private Sub Copy_Click()           '复制按钮的 Click 事件
     If Right(Dir2. Path, 1) = "\" Then
          strDestinFile = Dir2. Path & File1. FileName
     Else
          strDestinFile = Dir2. Path & "\" & File1. FileName
    End If
    FileCopy strSorceFile, strDestinFile
End Sub
Private Sub Dir1_Change()          '源文件保持与源目录列表框同步
    File1. Path = Dir1. Path
End Sub
  Private Sub Drive1_Change()      '源目录保持与源驱动器同步
    Dir1. Path = Drive1. Drive
End Sub
  Private Sub Exit_Click()         '目标目录保持与目标驱动器同步
     Dir2. Path = Drive2. Drive
End Sub
Private Sub File1_Click()          '选择源文件名赋给变量 strSorceFile
   If Right(Dir1. Path, 1) = "\" Then
        strSorceFile = Dir1. Path & File1. FileName
   Else
        strSorceFile = Dir1. Path & "\" & File1. FileName
  End If

End Sub
Private Sub Form_Load()            '设置启动时目标磁盘与目录
Drive2. Drive = "D:"
Dir2. Path = "D:\Vbprogs"
End Sub
```

8.4 通用对话框

在 Windows 操作系统及其应用程序中，有许多对话框其外观及操作都很相似，如“打开”对话框、“另存为”对话框、“颜色”对话框、“字体”对话框、“打印”对话框等。尽管利用 VB 所提供的文件管理控件可以方便地设计有关文件管理类的窗体，但是这些窗体无论是使用还是外观与 Windows 标准对话框都有一定的差距，而且设计这些窗体的代码编程量也非常之大。实际上，VB 也提供了用来制作这些标准对话框的控件，这就是通用对话框（CommonDialog）。

【案例八】 新建一个工程文件 vb8_3. vbp，该工程有一个窗体，在该窗体上使用通用对话框（CommonDialog）控件实现调用 Windows 标准对话框。通过调用 Windows 标准对话框实现对文本框文本字体、颜色的设置，并能将文本框中内容在打印机上打印。运行后的窗体如图 8-8 所示。

在图 8-8 中单击“打开”、“另存为”、“字体”、“颜色”和“打印”按钮，分别显示如图 8-9、图 8-10、图 8-11、图 8-12 和图 8-13 所示的对话框。

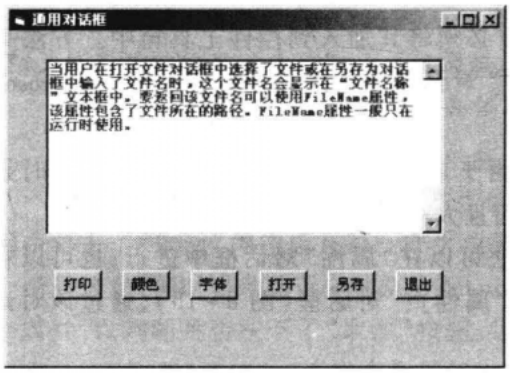


图 8-8 对话框应用示例效果图

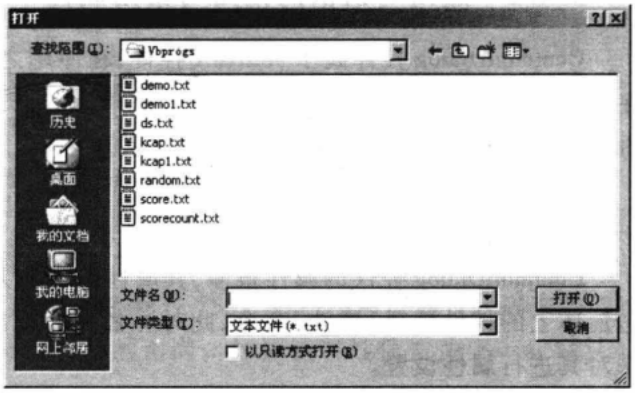


图 8-9 “打开”对话框

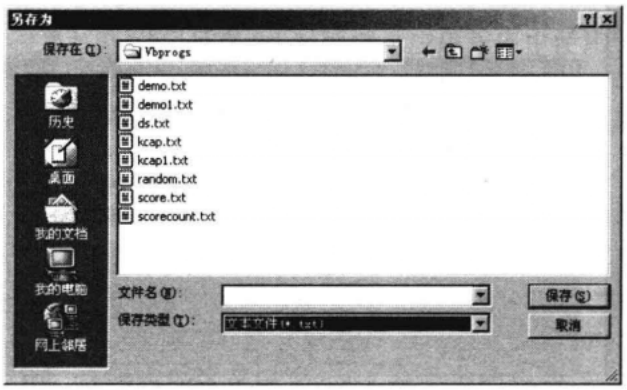


图 8-10 “另存为”对话框

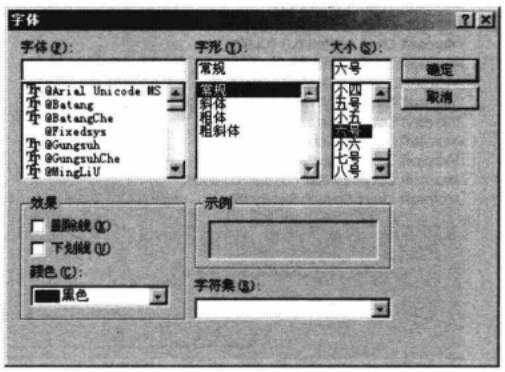


图 8-11 “字体”对话框

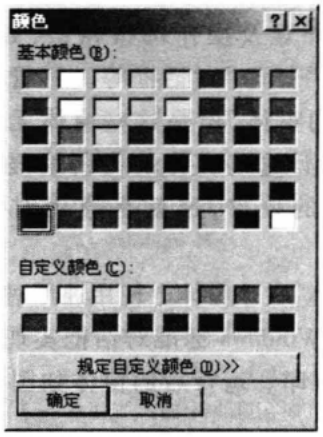


图 8-12 “颜色”对话框

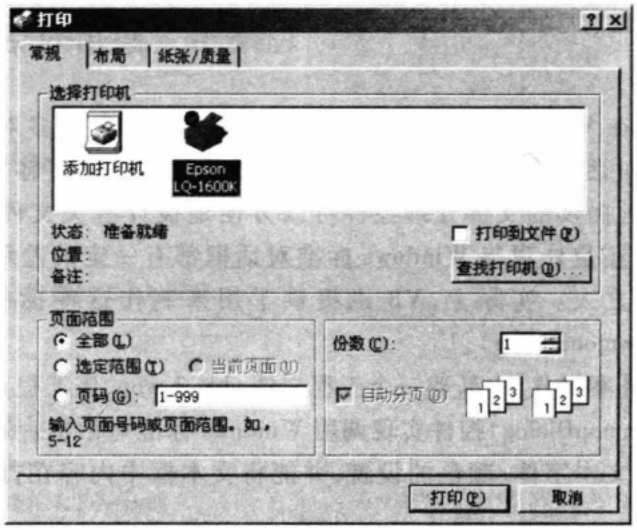


图 8-13 “打印”对话框

8.4.1 通用对话框的使用与常用属性

CommonDialog 控件属于 ActiveX 控件,在使用之前必须将其加入到控件工具箱中。方法是在控件工具箱上右击选择“部件”命令,弹出“部件”对话框(见图 8-14),选中“Microsoft Common Dialog Control 6.0”,最后单击“确定”按钮。

添加后,即可在窗体上使用通用对话框控件了。由于该控件仅在设计时可见,在运行时是不可见的,因此,可以将其放置在窗体的任意位置上,而且大小也不需要(也不能)调整。

CommonDialog 控件的属性很多,其属性设置的方法可以在“属性”对话框中进行;也可以右击窗体上的控件选择“属性”命令,此时屏幕上会弹出“属性页”对话框(图 8-15),通过该对话框对其进行属性设置。

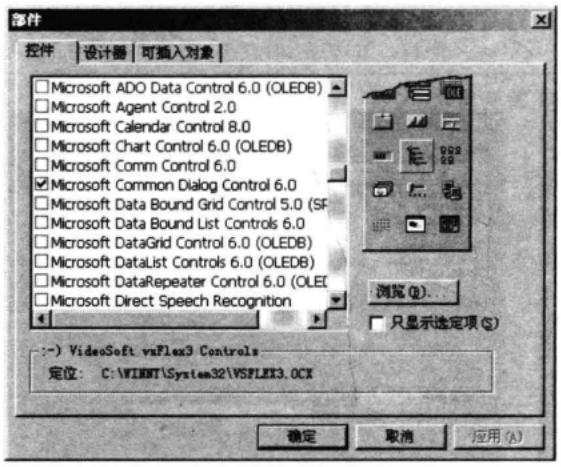


图 8-14 添加 CommonDialog 控件

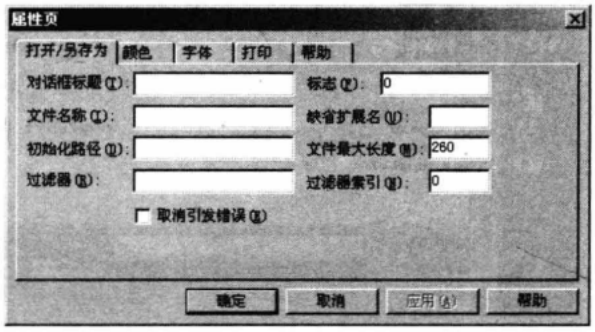


图 8-15 CommonDialog 控件“属性页”对话框

在该对话框中有 5 个选项卡,每个选项卡代表着利用通用对话框可以调用的不同的对话

框,它们有各自不同的属性,但有些属性是各个对话框所共有的,如下所示。

- Action 属性:通过在代码中设置不同的 Action 值可以产生不同的对话框。如果没有设置该属性(默认值为 0),运行时通用对话框不会产生任何信息,直到设置了该属性后才会出现相应的对话框。另外,VB 还提供了相应的方法来产生不同的对话框,即产生不同的对话框,要么设置 Action 属性,要么使用对应的方法。Action 属性值、方法与对话框的对应关系如表 8-5 所示。

表 8-5 Action 属性值、方法与对话框的对应关系表

Action 属性值	产生的对话框	对应的方法
0	无	
1	"打开"文件对话框	ShowOpen
2	"另存为"对话框	ShowSave
3	"颜色"对话框	ShowColor
4	"字体"对话框	ShowFont
5	"打印"对话框	ShowPrint
6	"帮助"对话框	ShowHelp

使用该控件所产生的对话框仅提供了人机交互界面,并不能实现真正的操作(如打开、另存为和打印等),要想实现这些具体的操作,必须进行相应的编程。

- DialogTitle 属性:产生对话框的标题文字,如不进行设置则会使用默认的标题。
- CancelError 属性:在每个所产生的对话框上都有"取消"按钮,该属性决定程序运行时,用户单击了"取消"按钮后是否会产生出错信息。该属性取 True 时,单击"取消"按钮会产生一个错误信息,同时自动将 Err 对象的 Number 属性值置为 32 755。程序中可以通过代码来访问这个属性,以判断是否按下了"取消"按钮,以决定程序的走向。如果该属性值置为 False(默认),则当单击"取消"按钮时,不会产生任何出错信息。

8.4.2 "打开"与"另存为"对话框

"打开"与"另存为"对话框是应用程序中使用较多的对话框,两种对话框的外观、设置及操作基本相同。当在代码中将通用对话框的 Action 属性设置为 1 或 2 时,即会产生"打开"或"另存为"对话框(图 8-9、图 8-10)。要使所产生的对话框起作用,还必须对其相关的属性进行一些必要的设置。

(1) 设置要打开的文件类型(Filter 属性)

通过 Filter 属性在"打开"对话框或"另存为"对话框中设置文件的类型。该属性的取值是一个字符串,由一组或多组文件类型表达式构成,每组代表一类文件。构成规则是:

类型说明字符串|类型通配表达式[|类型说明字符串|类型通配表达式]…

其中,"类型说明字符串"为对文件类型的说明,"类型通配表达式"表示需显示的文件类型,各组之间用"|"符号分开。例如,要显示所有扩展名为 .bas 或 .txt 的文件,可以用如下的属性设置:

```
cdlDisp. Filter = "VB 源程序( * . bas) | * . bas | 文本文件( * . txt) | * . txt"
```

运行时,在“打开”或“另存为”对话框中的“文件类型”下拉列表框中将显示该属性的值,显示的方法是每一类文件说明显示为一行(也就是列表框中的一个项目),每个项目都有一个FilterIndex 属性值,第一个项目为 1,第二个项目为 2,以此类推。FilterIndex 属性既可以在设计时设置,也可以在运行时返回。在设计时设置则规定了在初始运行时显示在列表框中的内容是第几项。如果在运行时设置,则表明用户进行了选择。例如,要在列表框中显示上述所列类型的第二类文件(所有的 . txt 文件),只要在代码中作如下设置:

```
cdlDisp. FilterIndex = 2
```

在“另存为”对话框中还有一个 DefaultExt 属性,用来设置默认的文件扩展名,其值是 1 ~ 3 个字符的字符串。当保存一个没有指定扩展名的文件时,将自动为文件添加默认的扩展名。

(2) 设置初始路径(InitDir 属性)

InitDir 属性主要用于设置“打开”或“另存为”对话框的初始路径,如果只对系统的当前路径操作,则不需对属性进行设置。初始路径在设计时或代码中均可进行设置。例如:

```
cdlDisp. InitDir = "D:\vbprogs"    '将对话框的初始路径设为 D 盘 vbprogs 文件夹
```

(3) 返回文件的名称(FileName 属性)

当用户在“打开”对话框中选择了文件或在“另存为”对话框中输入了文件名时,这个文件名会显示在“文件名称”文本框中。要返回该文件名可以使用 FileName 属性,该属性包含了文件所在的路径。FileName 属性一般只在运行时使用。

8.4.3 “字体”、“颜色”与“打印”对话框

除了“打开”与“另存为”对话框外,利用通用对话框还可以很方便地生成“字体”、“颜色”、“打印”等标准的 Windows 对话框。有关这些对话框的设置与使用方法见表 8-6 所示(各对话框的 Action 属性设置见表 8-5)。各对话框如图 8-11、图 8-12 和图 8-13 所示。

表 8-6 字体、颜色、打印对话框的主要属性设置

对话框	设计时设置的主要属性		运行时返回或设置的主要属性	
	名 称	意义及取值	名 称	意义及取值
字体	Flags	设置字体类型及对话框的格式(&H1, &H2, &H3, &H100)	FontName FontSize FontBold FontItalic	字体名称 字号大小 是否粗体 是否斜体
颜色	无	无	Color	所选择的颜色值
打印	Flags Min, Max	设置打印选项(详细内容请参见 VB 帮助) 页码范围的最小与最大值	Copies FromPage ToPage	打印份数 开始打印页 结束打印页

【例 8-7】 完成案例八利用通用对话框在一个窗体上单击“打开”、“另存为”、“字体”、

“颜色”和“打印”按钮，调出相应的 Windows 系列标准对话框实现对文本框文本字体、颜色的设置，并能将其在打印机上打印输出。

设计步骤如下：

① 窗体设计。在窗体上添加 1 个多行带滚动条的文本框、5 个命令按钮和 1 个通用对话框。

② 窗体及主要控件属性(“字体”、“颜色”和“打印”对话框的主要属性设置如表 8-7 所示)。

表 8-7 窗体及主要控件属性

控 件 名	属 性 名	设 置 值
“通用对话框”窗体	Name	frmExam_8_4
	Caption	通用对话框
通用对话框	Name	cdlDisp
文本框	Name	txtDisp
	Multiline	true
“打开”命令按钮	Name	cmdDisp
	Caption	打开
“另存”命令按钮	Name	cmdSave
	Caption	另存
“字体”命令按钮	Name	cmdFont
	Caption	字体
“颜色”命令按钮	Name	cmdColor
	Caption	颜色
“打印”命令按钮	Name	cmdPrint
	Caption	打印
“退出”命令按钮	Name	cmdExit
	Caption	退出

案例八相关事件的过程代码如下：

```
Dim strFname As String                          '存放文件名
Private Sub cmdDisp_Click()                     '"打开文件"事件
    Dim strLine As String
    cdlDisp. CancelError = True                 '单击"取消"按钮时引发错误
    On Error GoTo errhandler                    '启动错误处理程序
    With cdlDisp
        . Filter = "VB 源程序( * . bas) | * . bas|文本文件( * . txt) | * . txt"   '设置文件过滤器
        . FilterIndex = 2                       '默认过滤器为 2
        . InitDir = "D:\Vbprogs"                '设置初始路径
        . ShowOpen                              '显示打开对话框
    End With
```

```
strFname = cdlDisp.FileName                                    '返回文件名
txtDisp.Text = ""
Open strFname For Input As #1                                  '打开选定的文件
Do While Not EOF(1)
    Line Input #1, strLine                                     '读文件内容
    txtDisp.Text = txtDisp.Text + strLine + Chr(13) + Chr(10) '显示在文本框中
    Loop
    Close #1                                                   '关闭文件
    Exit Sub
    errhandler:
        errorDisp                                              '调用错误处理例程
  End Sub
```

```
  Private Sub cmdSave_Click()                                  '"另存为"按钮事件
      cdlDisp.CancelError = True
      On Error GoTo errhandler
      With cdlDisp
          .Filter = "VB 源程序(*.bas)|*.bas|文本文件(*.txt)|*.txt"
          .InitDir = "D:\Vbprogs"
          .ShowSave                                            '显示"另存为"对话框
      End With
      strFname = cdlDisp.FileName                              '返回输入的文件名
      If cdlDisp.FilterIndex = 1 Then
          strFname = Left(strFname, Len(strFname) - 3) + "bas"  '修改文件的扩展名
      End If
      Open strFname For Output As #2                           '以 output 方式打开文件
      Print #2, txtDisp.Text                                   '写文件
      Close #2
      Exit Sub
      errHandler:
        errorDisp
End Sub
```

```
Public Sub errorDisp()                                         '错误处理例程
  If Err.Number = 32755 Then                                   '进行错误处理
    MsgBox ("操作已取消", vbExclamation, "警告")                '显示警告信息
  End If
End Sub
```

```
Private Sub cmdFont_Click()                                    '"字体"按钮事件过程
    With cdlDisp
        .Flags = &H3 Or &H100                                  '设置"字体"对话框模式
        .ShowFont                                              '显示"字体"对话框
    End With
```

```
    With txtDisp                                    '返回相关属性并设置文本框文字属性
        . Font = cdlDisp. FontName                  '字体名称
        . FontBold = cdlDisp. FontBold              '是否粗体
        . FontItalic = cdlDisp. FontItalic          '是否斜体
        . FontStrikethru = cdlDisp. FontStrikethru  '是否加删除线
        . FontUnderline = cdlDisp. FontUnderline    '是否加下划线
        . ForeColor = cdlDisp. Color                '字体颜色
    End With
End Sub
```

```
Private Sub cmdColor_Click( )                       ' "颜色"按钮事件过程
    cdlDisp. Action = 3                             '显示"颜色"对话框
    txtDisp. ForeColor = cdlDisp. Color             '返回所选颜色并对文本框进行设置
End Sub
```

```
Private Sub cmdPrint_Click( )                       ' "打印"按钮的单击事件过程
    With cdlDisp                                    '设置"打印"对话框参数
        . Max = 999
        . Min = 1
        . FromPage = 1
        . ToPage = 999
    End With
    cdlDisp. Action = 5                             '显示"打印"对话框
    For i = 1 To cdlDisp. Copies                    '多份打印循环
        Printer. Print txtDisp. Text                '数据送往 Printer 对象
    Next i
    Printer. EndDoc                                 '打印内容
End Sub
```

程序运行后的效果如图 8-8 所示。

习 题 8

1. 什么是文件? 文件有哪些类型? 分别可使用什么方式访问?

2. 什么是顺序文件? 什么是随机文件?

3. 向一个顺序文件写数据时,可以使用 Output 或 Append 两种方法来打开。在写数据时,这两种方式有何不同?

4. 随机文件一般适用于存放什么类型的数据? 对随机文件的操作与顺序文件相比有何不同?

5. 有一个记录学生成绩的随机文件,每个记录包含如下内容:

(1) 学号(字符,8 个数字)。

(2) 姓名(字符,4 个汉字)。

(3) 政治(数值)。

(4) 高等数学(数值)。

(5) 计算机基础(数值)。

(6) 英语(数值)。

试定义一个记录型数据类型,并定义一个变量。将变量赋初值。(各字段内容自定)

6. 编写一段程序,用于将文本文件 text1. txt 合并到文本文件 text2. txt 的尾部。

7. 创建一个学生成绩录入窗体,学生成绩的结构如题5中所列。窗体的基本界面与图8-2相似,每个学生的成绩数据存放到 xscj. dat 中。

8. 如何使用文件管理的3个控件?怎样实现3个控件之间的同步?

9. 目录列表框与一般的列表框相比有什么不同?目录列表框中的 ListIndex 属性值是如何规定的?

10. 文件管理的3个控件都有 Change 事件,它们分别是如何引发的?有何作用?

11. 试编制一段代码实现如下功能:

(1) 判断当前目录下有多少个子目录。

(2) 将倒数第二个子目录变成当前目录。

(3) 显示该目录中的所有文件。

12. 如何返回用户在文件列表框中所选文件的文件名及所在路径?

13. 设计一个窗体将两个文本文件合并成第三个文件。需要进行合并的两个文本文件通过文件管理的3个控件由用户选择,同时提供合并后的文件保存,并能在窗体上显示合并后的文件内容(窗体界面及元素安排自行决定)。

14. 在案例七图8-6的界面上增加两个命令按钮,以实现对选定的源文件进行改名和删除功能。

15. 使用通用对话框可以实现哪些功能?如何生成各类对话框?

16. 完成本章的案例操作。

第9章　图形处理与多媒体应用

学习目标

了解 VB 中常用的图形控件的使用方法和图形程序的设计方法，能编制简单的动画程序；了解多媒体技术的特点，掌握 VB 中多媒体应用程序的编制方法。

随着计算机技术的发展，图形处理和多媒体已成为计算机应用的重要领域。VB 作为功能强大的 Windows 程序开发工具，提供了图形处理和多媒体应用的很多方法。本章简要介绍 VB 的图形处理和多媒体应用。

【案例九】 模拟卫星的运动。

在工程文件 vb9_2. vbp 中的窗体上有一个图片框，一个滚动条和 3 个命令按钮。程序运行后，单击“运行”按钮后，使小球围绕大球转动，并可以使用滚动条调节转动的速度；单击“暂停”按钮后，暂停小球的转动；按“结束”按钮结束程序。程序运行界面如图 9-1 所示。

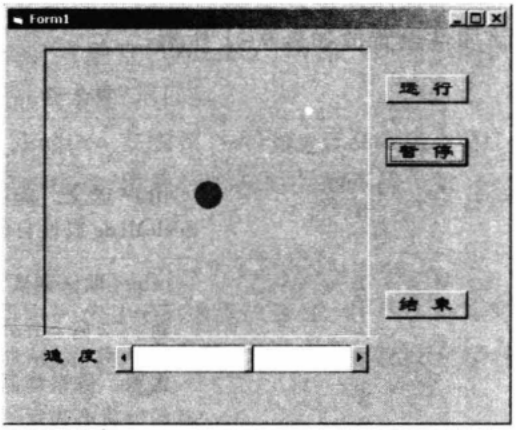


图 9-1　小球绕大球做圆周运动

9.1　图形处理

VB 为应用程序的使用提供了强大的图形功能，灵活运用这些功能，不仅可以使应用程序更具有吸引力，而且可以使应用程序更易于使用。

9.1.1　坐标系

要进行图形处理，首先要确定坐标系。本节描述的每一个图形操作（包括调整大小、移动和绘图）都要使用绘图区或容器的坐标系统。坐标系统是一个二维网格，可定义屏幕上、窗体中或其他容器中（如图片框或 Printer 对象）的位置。当使用窗体中的坐标时，可定义网格上的位置（x，y）：x 值是沿 X 轴点的位置，最左端是默认位置 0；y 值是沿 Y 轴点的位置，最上端是默认位置 0。该坐标系统的图标如图 9-2 所示。控件的 Left、Top 属性值就是左上角对应的（x，y）坐标。

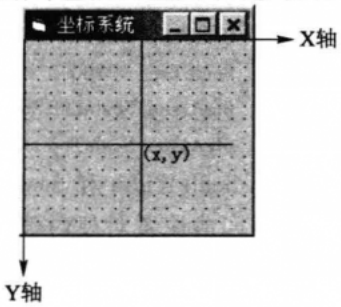


图 9-2　窗体的坐标系统

VB 坐标系统使用以下规则：

① 当移动控件或调整控件的大小时，使用控件容器的坐标系

统。如直接在窗体上绘制对象时,那么窗体就是容器。

② 所有的图形和 Print 方法使用容器的坐标系统。如在图片框里绘制控件的语句,使用的是控件的坐标系统。

③ 一些用来调整窗体大小或移动窗体的语句,用缇(twips)来表示窗体的位置和大小。

④ 任何容器的默认坐标是从左上角(0,0)坐标开始的。

沿坐标轴定义位置的测量单位,统称为刻度。每个窗体和图片框都有几个刻度属性(ScaleLeft、ScaleTop、ScaleWidth、ScaleHeight 和 ScaleMode)和一个方法(Scale),它们可用来定义坐标系统。对于 VB 中的对象,默认刻度把坐标(0,0)放置在对象的左上角。默认刻度单位为缇。

1. ScaleMode 属性

当使用图形方法或调整控件位置时,返回或设置一个值,该值指示对象坐标的度量单位。该属性设置值如表 9-1 所示。

表 9-1　ScaleMode 属性值及其描述

ScaleMode 设置值	描　述
0	用户定义。若直接设置了 ScaleWidth、ScaleHeight、ScaleTop 或 ScaleLeft,则 ScaleMode 属性自动设为 0
1	twips(默认刻度)。1 440 twips = 1 in
2	磅。72 磅 = 1 in
3	像素。像素是显示器或打印机分辨率的最小单位。每英寸里像素的数目由设备的分辨率决定
4	字符。打印时,一个字符有 1/6 in 高、1/12 in 宽
5	英寸(in)
6	毫米(mm)
7	厘米(cm)

设置 ScaleMode 的值后,VB 会重定义 ScaleWidth 和 ScaleHeight,使它们与新刻度保持一致。然后,将 ScaleTop 和 ScaleLeft 设置为 0。

2. ScaleLeft 和 ScaleTop 属性

返回或设置一个对象左上角水平(ScaleLeft)和垂直(ScaleTop)方向的坐标。下面的语句为当前窗体的左上角及 picArena 图片框的左上角设定数值。

```
ScaleLeft = 100
ScaleTop = 100
picArena. ScaleLeft = 100
picArena. ScaleTop = 100
```

3. ScaleWidth 和 ScaleHeight 属性

返回或设置对象内部(绘图区)的水平(ScaleWidth)或垂直(ScaleHeight)度量单位。例如:

```
ScaleWidth = 1000
ScaleHeight = 500
```

这两条语句定义当前窗体内部宽度的 1/1 000 为水平单位;当前窗体内部高度的 1/500 为

垂直单位。如果窗体的大小以后被调整,这些单位保持原状。

4. Scale 方法

将坐标系统重置到所选择的任意刻度。Scale 对运行时的图形语句以及控件位置的坐标系统都有影响。语法格式如下:

[object.]Scale (x1, y1) - (x2, y2)

其中,x1 和 y1 的值,决定了 ScaleLeft 和 ScaleTop 属性的设置值。两个 x 坐标之间的差值和两个 y 坐标之间的差值,分别决定了 ScaleWidth 和 ScaleHeight 属性的设置值。例如,假定要为一窗体设置坐标系统,而将两个端点设置为(100,100)和(200,200)。

```
Scale (100, 100) - (200, 200)
```

该语句定义窗体为 100 单位宽和 100 单位高。

5. CurrentX、CurrentY 属性

返回或设置下一次打印或绘图方法的水平(CurrentX)或垂直(CurrentY)坐标,即当前坐标。设计时不可用。用不同的图形方法时,CurrentX 和 CurrentY 的值不同:

用 Circle 方法时为圆、弧或椭圆的中心;用 Cls 方法、EndDoc 方法和 NewPage 方法时为(0,0);用 Line 方法时为线的终点;用 Print 方法时为下一个打印的位置;用 Pset 方法时为画点的位置。

9.1.2 颜色的使用

在 VB 中,控件和图形的颜色属性中的每种颜色都由一个 Long 型整数表示,并且在指定颜色的所有上下文中,该数值的意义相同。在运行时通常使用 RGB 函数、QBColor 函数或直接输入一种颜色值来确定颜色。

1. RGB 函数

它返回代表 RGB 色彩值的长整型数值,可用来指定任何颜色。语法格式如下:

```
RGB(red, green, blue)
```

其中,参数 red、green、blue 分别为红、绿、蓝三基色值,其值均在 0 ~ 255 之间,0 表示亮度最低,而 255 表示亮度最高。如:

```
MyObject.Color = RGB(255, 0, 0)          '设置 MyObject(红色)
```

表 9-2 给出了一些常见的标准颜色,以及这些颜色的红、绿、蓝三基色的成分。

表 9-2 常见的标准颜色

颜　色	red 值	green 值	blue 值
黑色	0	0	0
蓝色	0	0	255
绿色	0	255	0
青色	0	255	255
红色	255	0	0
洋红色	255	0	255
黄色	255	255	0
白色	255	255	255

2. QBColor 函数

该函数返回一个长整型,用来表示所对应颜色值的 RGB 颜色码。语法格式如下:

QBColor(color)

其中,参数 color 是一个界于 0 ~ 15 的整型数。color 参数的设置如表 9-3 所示。

表 9-3　常见的标准颜色及内部颜色常量

内部颜色常量	值	颜　色	内部颜色常量	值	颜　色
vbBlack	0	黑色	vbGray	8	灰色
vbBlue	1	蓝色	vbLightBlue	9	亮蓝色
vbGreen	2	绿色	vbLightGreen	10	亮绿色
vbCyan	3	青色	vbLightCyan	11	亮青色
vbRed	4	红色	vbLightRed	12	亮红色
vbMagenta	5	洋红色	vbLightMagenta	13	亮洋红色
vbYellow	6	黄色	vbLightYellow	14	亮黄色
vbWhite	7	白色	vbBrightWhite	15	亮白色

3. 设置颜色值

VB 中许多控件的 BackColor、ForeColor 属性(决定它的背景、前景颜色),其颜色可直接使用表示颜色的内部常量(表 9-3)。如将背景色定义为 vbRed:

```
BackColor = vbRed
```

颜色属性也可直接使用颜色设置值。多数情况下,用十六进制数表示这些数值更简单。正常的 RGB 颜色的有效范围是从 0 ~ 16 777 215(&HFFFFFF)。每种颜色的设置值(属性或参数)都是一个 4 字节的整数。对于这个范围内的数,其高字节都是 0,而其他个字节,从最低字节到第 3 个字节,分别定义了红、绿、蓝三种颜色的值。红、绿、蓝三种成分都是用 0 ~ 255(&HFF)之间的数表示。因此,可以用十六进制数按照下述语法来指定颜色:

```
&HBBGGRR
```

其中,BB 指定蓝颜色的值,GG 指定绿颜色的值,RR 指定红颜色的值。每个数段都是两位十六进制数,即从 00 ~ FF。如:

```
BackColor = &H808080                    '设置对象的背景色为灰色
```

9.1.3　图形控件的使用

为了在应用程序中创作图形,VB 提供了 3 种控件:Line 控件、Image 控件和 Shape 控件,此外还提供 PictureBox 控件以显示图形。这几种控件的用法简述如下。

1. Line 控件

Line 控件用以显示水平线、垂直线或者对角线。线条的特性由下面几个属性决定。

(1) BorderColor 属性

返回或设置对象的边框颜色。可使用调色板或在代码中使用 RGB 或 QBColor 函数指定颜色。默认为黑色。

(2) BorderStyle 属性

返回或设置对象的边框样式(图 9-3)。Line 及 Shape 控件的 BorderStyle 属性设置值见表 9-4 所示。

(3) BorderWidth 属性

返回或设置控件边框的宽度。其值从 1~8 192。默认为 1。

(4) x1、y1、x2、y2 属性

返回或设置 Line 控件的起始点(x1,y1)和终止点(x2,y2)的坐标。水平坐标是 x1 和 x2;

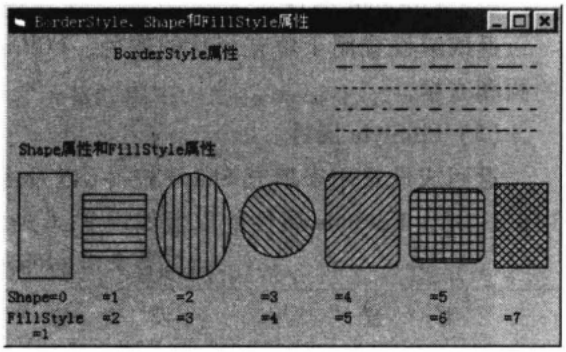


图 9-3 BorderStyle、Shape 和 FillStyle 属性示例

表 9-4 BorderStyle 属性值及其描述

常 量	设 置 值	描 述
vbTransparent	0	透明
vbBSSolid	1	实线(默认值)。边框处于形状边缘的中心
vbBSDash	2	虚线
vbBSDot	3	点线
vbBSDashDot	4	点画线
vbBSDashDotDot	5	双点画线
vbBSInsideSolid	6	内收实线。边框的外边界就是形状的外边缘

垂直坐标是 y1 和 y2。运行时不能使用 Move 方法移动 Line 控件,但可以通过改变 x1、x2、y1 和 y2 属性来移动它或者调整它的大小。

2. Shape 控件

Shape 控件可以显示矩形、正方形、椭圆形、圆形、圆角矩形或者圆角正方形。该控件的特性由下面几个属性决定:

(1) Shape 属性

返回或设置一个整数值,用该值指示一个 Shape 控件的外观(图 9-3)。Shape 属性取值如表 9-5 所示。

表 9-5 Shape 属性值及其描述

常 量	设 置 值	描 述
vbShapeRectangle	0	矩形(默认值)
vbShapeSquare	1	正方形
vbShapeOval	2	椭圆形
vbShapeOval	3	圆形
vbShapeRoundedRectangle	4	圆角矩形
vbShapeRoundedSquare	5	圆角正方形

（2）FillColor 属性

返回或设置用于填充形状的颜色。FillColor 也可以用来填充由 Circle 和 Line 图形方法生成的圆和方框。

（3）FillStyle 属性

返回或设置用来填充 Shape 控件（以及由 Circle 和 Line 图形方法生成）的圆和方框的模式（图 9-3）。FillStyle 属性的取值如表 9-6 所示。

表 9-6　FillStyle 属性值及其描述

常　量	设 置 值	描　述
vbFSSolid	0	实线
vbFSTransparent	1	透明（默认值）
vbHorizontalLine	2	水平直线
vbVerticalLine	3	垂直直线
vbUpwardDiagonal	4	上斜对角线
vbDownwardDiagonal	5	下斜对角线
vbCross	6	十字线
vbDiagonalCross	7	交叉对角线

如果 FillStyle 属性设置为 1（透明），则忽略 FillColor 属性。下面的例子是在单击鼠标时，用随机的 FillColor 和 FillStyle 属性的设置值在窗体中显示一个圆。

```
Private Sub Form_MouseDown (Button As Integer, Shift As Integer, x As Single, y As Single)
    FillColor = RGB(Rnd * 15)          '选择随机的 FillColor
    FillStyle = Int(Rnd * 8)           '选择随机的 FillStyle
    Circle (x, y), 250                 '画一个圆
End Sub
```

3. Image 控件

Image 控件可以显示来自位图、图标或元文件的图形，也可以显示 JPEG 或 GIF 文件。

（1）Picture 属性

返回或设置控件中要显示的图片。设计时可从属性窗口中加载图片，在运行时，使用 LoadPicture 函数来加载位图（.bmp）、图标（.ico）或元文件（.wmf）。如：

```
Image1. Picture = LoadPicture("C:\Windows\Circles. bmp")
```

如果在设计时设置 Picture 属性，图片被保存起来并与窗体同时加载。如果创建可执行文件，该文件中包含该图像。如果在运行时加载图片，该图片不和应用程序一起保存。

（2）Stretch 属性

返回或设置一个值（Boolean），该值用来指定一个图形是否要调整大小，以适应 Image 控件的大小。True 表示图形拉伸适应控件的大小，False（默认值）表示不用拉伸。

4. PictureBox 控件

PictureBox 控件可以显示来自位图、图标或者来自元文件、JPEG 及 GIF 文件的图形。如果

控件不足以显示整幅图像,则裁剪图像以适应控件的大小。

(1) AutoSize 属性

设置 PictureBox 控件是否能够自动调整大小以显示整幅图形。True 表示允许。

(2) Picture 属性(同 Image 控件)

9.1.4 常用的绘图方法

除了使用图形控件创建图形之外,VB 还提供了创建图形的一些方法。这些方法适用于窗体和图片框。

1. Cls 方法

Cls 方法清除图形和打印语句在运行时所产生的文本和图形,而设计时在 Form 中使用 Picture 属性设置的背景位图和放置的控件不受 Cls 方法影响。

调用 Cls 之后,对象的 CurrentX 和 CurrentY 属性复位为 0。

2. PSet 方法

将对象上的点(x, y)设置为指定颜色,即在(x,y)处画点。语法格式为:

```
object. PSet [Step] (x, y), [color]
```

其中各参数的含义如下。

- Step:可选。(x,y)表示与当前坐标的相对位置。
- color:可选。Long 型,为该点指定的 RGB 颜色。若省略,则使用当前的 ForeColor 属性值。

PSet 方法常用来画曲线。例如:

```
'单击窗体时画一条放大的正弦曲线
Private  Sub Form_Click()
    Dim  x, y
    For x = 0 to 7200
        PSet (x, 800 * Sin(x * 3.14159 / 1800 ) + 1200) , RGB(255, 0 , 0)
    Next  x
End Sub
```

要用 PSet 方法清除单一像素,可规定该像素的坐标,并用 BackColor 属性设置作为 color 参数。

3. Point 方法

按照长整数,返回指定点(x, y)的(RGB)颜色。语法格式为:

```
object. Point(x, y)
```

4. Line 方法

Line 方法用于绘制直线或矩形,其语法格式如下:

```
[ <对象名>. Line  [Step](x1, y1) - [Step] (x2,y2)  [,[ <颜色>][,B[F]]
```

<对象名>:要绘制直线或矩形的容器对象名称,如窗体、图片框等,默认为当前窗体。

(x1, y1):可选项,起点坐标。如果省略,图形起始于由 CurrentX 和 CurrentY 指示的位置。

(x2, y2):终点坐标。

Step:可选项,当在(x1, y1)前出现 Step 时,表示(x1, y1)是相对于由 CurrentX 和 CurrentY 指示的位置;当在(x2, y2)前出现时,表示(x2, y2)为相对于图形起点的终点坐标。

<颜色>:直线或矩形的颜色,如果省略,则使用 ForeColor 属性的值作为直线或矩形的颜色。

B:可选项,如果选择了 B,则以(x1, y1)、(x2, y2)为对角坐标画直线(当(x2, y2)后只有一个“,”时)或矩形(当(x2, y2)后有两个“,”时)。

F:可选项,如果使用了 B 参数后再选择 F 参数,则规定矩形以矩形边框的颜色填充。如果不使用 F 参数只使用 B 参数,则矩形用当前容器对象的 FillColor 和 FillStyle 填充。FillStyle 的默认值为 1 - Transparent(透明)。不能只选择 F 参数而不选择 B 参数。

画连续直线时,前一条直线的终点就是后一条直线的起点。线的宽度取决于 DrawWidth 属性值。执行 Line 方法后,当前坐标(CurrentX 和 CurrentY 属性)被设置在终点坐标(x2, y2)处。

5. Circle 方法

Circle 方法可画出圆形和椭圆形的各种形状以及圆弧。

(1) 画圆

画圆时,Circle 方法的语法格式为:

```
[object.]Circle[Step](x,y),radius[,color]
```

其中各参数的含义如下。

- (x,y):圆心的坐标。带关键字 Step 时,表示与当前坐标的相对位置。
- radius:圆的半径。
- color:可选。画圆时用的 RGB 颜色。如果它被省略,则使用 ForeColor 属性值。例如,下面语句将画出一个以(1 000,1 200)为圆心、850 为半径的圆:

```
Circle (1000, 1200), 850
```

可使用绘图区的标尺属性,将该圆的圆心置于窗体的中心处。例如下列语句:

```
Circle ((ScaleWidth + ScaleLeft) / 2, (ScaleHeight + ScaleTop) / 2), ScaleWidth / 4
```

【例 9-1】 多彩地毯演示程序。

新建工程文件 vb9_1.vbp,窗体命名为“frmExam_9_1”,在窗体的单击事件中加入调用 CircleDemo 过程命令。CircleDemo 过程围绕窗体中心画出一系列虚线圆,多次单击画出的这些圆,使其看起来像一个正在编织的圆形地毯(图 9-4)。

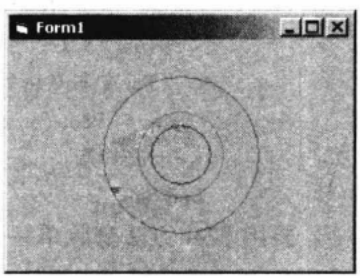


图 9-4　圆形地毯

代码清单如下:

```
Private Sub Form_Click()
    CircleDemo
End Sub

Sub CircleDemo ()
    Dim Radius
    R = 255 * Rnd                          '将红色设置为随机数
    G = 255 * Rnd                          '将绿色设置为随机数
    B = 255 * Rnd                          '将蓝色设置为随机数
    XPos = ScaleWidth / 2                  '将 x 坐标设置在窗体中间
    YPos = ScaleHeight / 2                 '将 y 坐标设置在窗体中间
    Radius = ((YPos * 0.9) + 1) * Rnd      '将半径设置在窗体高度的 0 到 50% 之间
    Circle (XPos, YPos), Radius, RGB(R, G, B)  '用随机颜色画圆
End Sub
```

(2) 画圆弧

画圆弧时,Circle 方法的语法格式为:

[object.]Circle [Step](x, y), radius, [color], start, end

其中,start 和 end 是以弧度为单位指定圆弧的起始角和终止角。如果 start 或 end 是负数,将画一条连接圆心到起点和终点的直线,相当于画一个饼。

例如,下面的过程将画出一个饼。

```
Private Sub Form_Click ()
    Const PI = 3.14159265
    Circle (1500, 1500), 1000, , -PI / 2, -PI / 3
End Sub
```

(3) 画椭圆

画椭圆时,Circle 方法的语法格式为:

[object.]Circle [Step](x, y), radius, [color], [start], [end] [, aspect]

其中各参数的含义如下。

- start 和 end:可选,用于指定椭圆的起止位置。如果想省掉参数,要加逗号。例如,如果含有 radius 和 aspect 参数,省掉了 color、start、end 3 个参数,则必须加上 4 个连续的逗号,表明它们被省掉了。例如:

```
Circle (1000, 1000), 500, , , , 2
```

- aspect:指定宽高比。默认为 1。当它小于 1 时,椭圆长轴在水平方向上,短轴在垂直方向上;当它大于 1 时,椭圆长轴在垂直方向上,短轴在水平方向上。

下列过程说明不同的 aspect 参数值,用 Circle 方法画出的不同椭圆:

```
Private Sub Form_Click ()
    FillStyle = 0
    Circle (600, 1000), 800, , , , 3          '画一个实心椭圆
    FillStyle = 1
    Circle (1800, 1000), 800, , , , 1 / 3      '画一个空心椭圆
End Sub
```

6. PaintPicture 方法

用于在 Form、PictureBox 或 Printer 上绘制图形文件(.bmp、.wmf、.emf、.cur、.ico 或 .dib)的内容,复制时可实现图形翻转。语法格式为:

object.PaintPicture picture, x1, y1, width1, height1, x2, y2, width2, height2, opcode

其中各参数的含义如下。

- picture:要绘制到 object 上的图形源。Form 或 PictureBox 的 Picture 属性。
- x1,y1:指定在 object 上绘制 picture 的目标左上角坐标(x 轴和 y 轴)。
- width1:可选。指定 picture 的目标宽度。
- height1:可选。指定 picture 的目标高度。
- x2,y2:可选。指定 picture 内剪贴区的左上角坐标(x 轴和 y 轴)。如果该参数省略,则默认为(0,0)。
- width2:可选。指定 picture 内剪贴区的源宽度。如果该参数省略,则使用整个源宽度。如果为负,则水平方向翻转图形。

- height2：可选。指定 picture 内剪贴区的源高度。如果该参数省略，则使用整个源高度。如果为负，则垂直方向翻转图形。如果 width2 和 height2 均为负，则两个方向翻转图形。
- opcode：可选。指定在将 picture 绘制到 object 上时对源像素与目标像素组合的方式：逻辑与、或、非、异或等（如 vbMergeCopy 或 vbSrcAnd 操作符）。

只复制图像，可用下面形式的代码：

```
Picture2. PaintPicture Picture1. Picture, 0, 0, _
    Picture1. ScaleWidth, Picture1. ScaleHeight, 0, 0, _
    Picture1. ScaleWidth, Picture1. ScaleHeight, &HCC0020
```

复制时图像目标水平翻转，可用下面形式的代码：

```
Picture2. PaintPicture Picture1. Picture, 0, 0, _
    Picture1. ScaleWidth, Picture1. ScaleHeight, Picture1. ScaleWidth, _
    0, -Picture1. ScaleWidth, Picture1. ScaleHeight, &HCC0020
```

复制时目标图像两个方向都翻转，可用下面形式的代码：

```
Picture2. PaintPicture Picture1. Picture, 0, 0, _
    Picture1. ScaleWidth, Picture1. ScaleHeight, Picture1. ScaleWidth, _
    Picture1. ScaleHeight, -Picture1. ScaleWidth, -Picture1. ScaleHeight, &HCC0020
```

7. Print 方法

用于在窗体或图片框上显示文本，语法格式为：

```
[object.]Print [outputlist] [{ ; | , }]
```

参数 object 是可选项，如果被省略，则 Print 方法将应用于当前窗体。Outputlist 为输出项表。

【例 9-2】 设计案例九中窗体及各控件属性并编写图 9-1 中命令按钮的 Click 事件代码及计时器 Timer 事件代码。

解题步骤：

第一步：建立界面并设置控件属性（表 9-7）。

表 9-7　工程 vb9_1 中控件属性

控件名	属性名	设置值
按钮 1	Caption	运行
	Name	Cmd_run
按钮 2	Caption	暂停
	Name	Cmd_pause
按钮 3	Caption	退出
	Name	Cmd_exit
滚动条	Name	HScroll1
图片框	Name	Picture1
Timer1	Name	Timer1

第二步：在案例九图 9-1 中各控件的事件代码如下。

```
Option Explicit
Dim c As Single, r As Single        'r 为小球到大球的球心的距离,c 为小球的角度
```

```
Dim x As Single, y As Single        'x,y 为小球移动时的圆心
Dim st As Single
```

```
Private Sub CMD_EXIT_Click( )
  Unload Me
End Sub
```

```
Private Sub CMD_PAUSE_Click( )
   Timer1. Enabled = False
End Sub
```

```
Private Sub CMD_RUN_Click( )
   Timer1. Enabled = True
End Sub
```

```
Private Sub Form_Load( )
   r = 20
   c = 0
   st = 0.063
   HScroll1. Min = 1                          '设置最小值
   HScroll1. Max = 100                        '设置最大值
   Picture1. Scale ( -23, 23) -(23, -23) '为了保证大球的圆心坐标(0,0)位于图片框的中心而改变容
                                              '器坐标系为常规坐标系
End Sub
```

```
Private Sub HScroll1_Change( )
   Timer1. Interval = 200 - HScroll1. Value   '根据滚动条的数值设置时间间隔,速度越快,间隔越小
End Sub
```

```
Private Sub Picture1_Paint( )
   Picture1. FillColor = QBColor(12)
   Picture1. ForeColor = QBColor(12)
   Picture1. Circle (0, 0), 2
   x = Cos(c) * r
   y = Sin(c) * r
   Picture1. FillColor = QBColor(14)
   Picture1. ForeColor = QBColor(14)
   Picture1. Circle (x, y), 0.5
   c = c + st
   If c > = 2 * 3.14159 Then
     c = c Mod (2 * 3.14159)
   End If
End Sub
```

```
Private Sub Timer1_Timer( )
     Picture1. Refresh '重画图片框
End Sub
```

解析：

① Picture1. Scale (-23, 23)-(23, -23) 为了保证大球的圆心坐标(0, 0)位于图片框的中心而改变容器坐标系为常规坐标系。

② 计时器控件作用是以一定的时间间隔激发计时器事件(Timer)执行相应程序代码。当计时器控件的 Enabled 属性为 True 时，计时器计时开始，当为 False 时计时器计时暂停，所以要使小球运动暂停可将 Enabled 属性值设置为 False，要使小球运行，则设置该属性值为 True。

9.1.5　简单动画设计

要使屏幕上的图形动起来，有两种简单的方法：图形方法和使用图形控件。同时为了使图形运动，需定时画图或显示图形，因而要用到 VB 的定时器控件。

1. 图形方法

对使用图形方法建立的动画，其实现方法是先绘制图形；用背景色重画该图形，即把图形擦除；接着移动位置再重复绘制图形，从而实现动画过程。

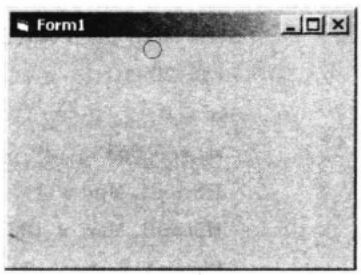


图 9-5　红色小球反弹

【例 9-3】　动画之一，即小球的跳动。要求在窗体中显示一个运动的红色小球，如果到达窗体边界则反弹(图 9-5)。

在新建窗体(命名为“frmExam_9_3”)中加一个定时器控件，工程名为 vb9_3. vbp。

所需代码如下：

```
Dim stepX, stepY As Integer                    '用于存储当前图形的位置和图形移动的步长值
Dim x, y As Integer
Private Sub Form_Load( )
    stepX = 20                                 '设置移动步长,起始位置
    stepY = 20
    x = 50
    y = 50
    Timer1. Enabled = True                      '定时器初始化
    Timer1. Interval = 10
End Sub
Private Sub Timer1_Timer( )                    '定时执行的动作
    Circle (x, y), 100, BackColor               '用背景色画图——擦除球
    If  x + 50 + stepX > ScaleWidth  Or     x - 50 + stepX < ScaleLeft  Then  '判断是否到边界
        stepX = -stepX                         '是,改变移动方向
    End If
      If y + 50 + stepY > ScaleHeight Or y - 50 + stepY < ScaleTop Then
         stepY = - stepY
    End If
      x = x + stepX                            '改变坐标位置
      y = y + stepY
      Circle (x, y), 100, RGB(255, 0, 0)       '用红色画球,此处可画其他图形
End Sub
```

2. 使用图形控件

图形控件有 Line 和 Shape 等,Shape 的属性值有 6 个值,当值为 0 时,Shape 控件的形状为矩形。在 Windows 应用程序中常用的进度条就可以用 VB 的图形控件 Shape 实现。

【例 9-4】 动画之二——电影字幕。

要求:单击"效果"按钮时,开始出现电影字幕的效果;在标签文字移动的过程中"效果"按钮不可选,移动结束后"效果"按钮恢复可选状态。在字幕移动过程中用一个"进度条"做标记,记录字幕移出的比例(图 9-6)。

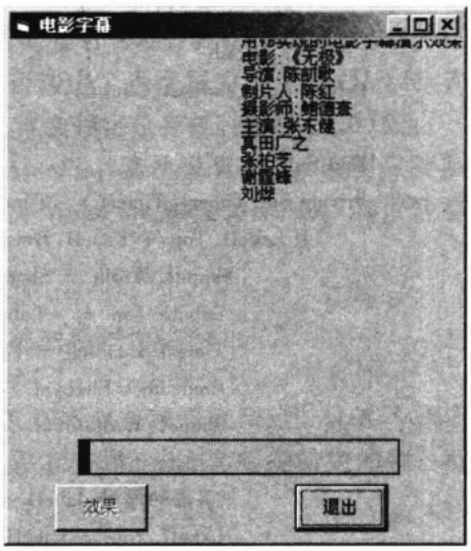


图 9-6 进度条标记电影字幕

操作步骤:

① 在窗体上放置控件并设置控件的属性。

新建一个工程文件 vb9_4.vbp,在窗体(窗体名命名为"frmExam_9_4")中放置一个 Label1 控件、两个 Commandbutton 控件和两个 Shape 控件,再放置一个定时器(Timer1)控件。

② 程序清单如下:

```
Option Explicit
Dim step As Integer
Private Sub CmdEffect_Click()
        CmdEffect.Enabled = False
        step = 30
        Label1.AutoSize = True
        Label1.Left = Form1.Width / 2
        Label1.Top = 20
        Label1.Caption = "用 VB 实现的电影字幕演示效果" + Chr$(13) + "电影:《无极》"_
        + Chr$(13) + "导演:陈凯歌" + Chr$(13) + "制片人:陈红" + Chr$(13)
        + "摄影师:鲍德熹"_ + Chr$(13)
        + "主演:张东健" + Chr$(13) + "真田广之" + Chr$(13) + "张柏芝" + Chr$(13)_
        + "谢霆锋" + Chr$(13) + "刘烨" + Chr$(13)
        '定义进度条 1 为蓝色并填充
        Shape1.BorderColor = RGB(0, 0, 150)
        Shape1.FillColor = Shape1.BorderColor
        Shape1.FillStyle = 0
        '定义进度条 1 的大小
        Shape1.Height = Shape2.Height - 2
        Shape1.Top = Shape2.Top + 1
        Shape1.Left = Shape2.Left
        Shape1.Width = 1
        If step < 0 Then
                step = - step
        End If
        If step = 0 Then
```

```
                step = 30
            End If
            Timer1.Interval = step

            Timer1.Enabled = True
            End Sub
    Private Sub CmdExit_Click()
            End
    End Sub
    Private Sub Timer1_Timer()   'Timer 事件
        If Label1.Top + Label1.Height < step Then
                Shape1.Width = Shape2.Width
                Label1.Top = -Label1.Height
                Timer1.Enabled = False
                CmdEffect.Enabled = True
                Shape1.Width = 1
        Else
                '字幕的滚动,label1.top 为负数即顶端移出屏幕
                Label1.Top = Label1.Top - step
                '实现进度条宽度变化效果
                Shape1.Width = Shape2.Width * (-Label1.Top / Label1.Height)
        End If
    End Sub
```

③ 解析：

动画效果和进度条效果都要通过定时器控件的 Timer 事件触发来实现。

把需要移动的字幕以多行方式用代码写在标签框的 Caption 属性中，将标签框的 Top 逐步向窗体 Y 坐标轴的负方向移动，即可得到电影字幕效果。用两个 Shape 控件组合使用来模拟进度条效果，运行时两个控件发生重叠，其中一个填充颜色，并且宽度逐渐加大，直到与另一个完全重叠为止。

9.2　多媒体应用

在 VB 的控件工具箱中并没有与多媒体有关的控件，多媒体应用控件是以 ActiveX 控件（又称“部件”）提供的。VB 中与多媒体有关的 ActiveX 控件使用较多的有两个：一个是 Animation 控件，另一个是“媒体控制器（MCI）”控件。



图 9-7　CD 播放器运行界面

【案例十】 设计 CD 播放器。在较早的 Windows 操作系统（如 Windows 98、Windows 2000 等）中有一个 CD 播放器，用于播放 CD 音乐，本案例将完成模拟 CD 播放器的程序设计。模拟 CD 播放器的界面如图 9-7 所示。

9.2.1 Animation 控件

VB 提供的 ActiveX 控件 Animation 能播放无声的 AVI 文件。要使用它需通过〖工程〗→〖部件〗向工具箱添加“Microsoft Windows Common Controls - 2 6.0”。该控件的主要属性与方法如下。

- AutoPlay 属性:返回或设置将 . avi 文件加载到控件时是否自动开始播放该文件。
- Center 属性:在 Animation 控件内确定 . avi 文件是否居中。当设置为 True(默认)时,根据图像的大小,在控件中心显示该文件;当设置为 False 时,该文件定位在控件内的(0,0)处。
- Open 方法:打开一个要播放的 . avi 文件。如果 AutoPlay 属性设置为 True,则只要加载该文件,就开始不断重复播放它,直到关闭该文件或设置 AutoPlay 属性为 False。其语法格式为:

```
object. Open  aviFileName
```

- Play 方法:在 Animation 控件中播放 . avi 文件。其语法格式为:

```
object. Play [ = repeat, start, end]
```

其中,repeat 为指定重复播放的次数。默认值为 -1,它使重复播放次数不受限定。start 为指定开始的帧。默认值是 0,表示在第一帧上开始播放,最大值是 65 535。end 为指定结束的帧。默认值是 -1,表示上一次播放的帧,最大值是 65 535。

- Stop 方法:在 Animation 控件中终止那些用 Play 方法启动的动画。
- Close 方法:在 Animation 控件中关闭打开的文件。

【例 9-5】 播放无声动画程序(设计时的窗体及控件如图 9-8 所示)。

操作步骤:

① 在窗体上放置控件并设置控件的属性。

新建一个工程文件 vb9_5. vbp,在窗体(窗体名命名为“frmExam_9_5”)中放置一个通用对话框和 Animation 控件以及两个命令按钮(Play、Stop)。

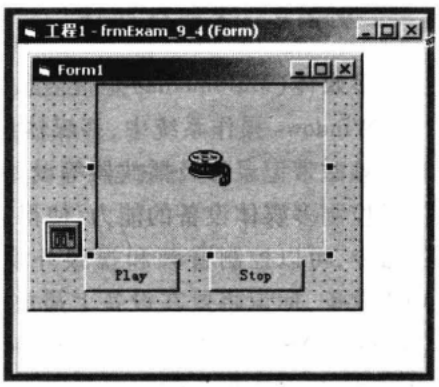


图 9-8 播放无声动画窗体及控件

② 窗体及控件的属性如表 9-8 所示。

表 9-8 窗体及控件属性的设置

控 件 名	属 性 名	设 置 值
Animation	Name	anmavi
CommonDialog	Name	dlgopen
命令按钮 1	Name	cmdPlay
	Caption	Play
命令按钮 2	Name	cmdStop
	Caption	Stop

③ 命令按钮的代码如下：

```
Private Sub cmdPlay_Click( )
    dlgOpen. Filter = "avi 文件 ( * . avi) | * . avi"
    dlgOpen. ShowOpen
    anmAVI. Open dlgOpen. FileName      '无休止地播放该文件
    anmAVI. Play
End Sub

Private Sub cmdStop_Click( )
        anmAVI. Stop                    '停止播放
End Sub
```

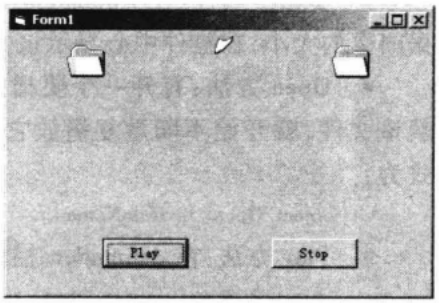


图 9-9　播放无声动画的效果

④ 单击“播放”命令按钮会出现一个“打开”对话框，在对话框中选择 FileCopy. avi 文件(在 VB 系统中附带有若干个无声的“. avi”文件，存放在 VB 系统文件夹\common\graphics\videos 下)会出现如图 9-9 所示的画面。

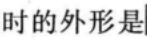
注意：如果试图用动画控件播放有声的“. avi”文件，系统将给出代码为 35752 的错误信息。

*9.2.2　多媒体概述

多媒体(Multimedia)是指信息的载体是文字、声音、图形和图像等形式的组合。

Windows 操作系统中，多媒体设备指声卡、MIDI 序列发生器、CD-ROM 驱动器、媒体播放器和视频磁带记录器及播放器等设备，MCI(Media Control Interface，媒体控制接口)为应用程序提供了控制多媒体设备的能力，使得应用程序与设备无关。该接口包含一组高层次的独立于设备的命令，可以控制音频和视频外设。首先发送的 MCI 命令就是 Open。这条命令打开指定的 MCI 设备，标识将要在设备上播放或记录的文件。设备打开后，可以发送任何其他的 MCI 命令(如 Prev、Next 和 Pause 等)。Close 命令是向设备发送的最后一条 MCI 命令，它释放所用的资源、关闭与设备相关的数据文件。

9.2.3　多媒体控件 MMControl

VB 中提供的 ActiveX 控件 Multimedia MCI(MMControl)用于管理媒体控制接口设备上的多媒体文件的记录与回放，可播放当前流行的多媒体格式文件，包括 . wav 文件、. avi 文件和音频 CD。为使用它，应向工具箱中添加“Microsoft Multimedia Control 6. 0”(即 MMControl 控件)。将 MMControl 控件加入到窗体时的外形是，它上面的一排按钮从左到右依次定义为 Prev、Next、Play、Pause、Back、Step、Stop、Record 和 Eject(即前一个、下一个、播放、暂停、向后步进、向前步进、停止、录制和弹出)。

1. MMControl 控件的主要属性

(1) AutoEnable 属性

指定 MMControl 控件是否能够自动启动或关闭控件中的某个按钮。如果该属性被设置为 True，MMControl 控件就启用指定 MCI 设备类型在当前模式下所支持的全部按钮。

(2) ButtonEnabled 属性

指定是否启用或禁用控件中的某个按钮。该属性可以被 Enabled 和 AutoEnable 属性所替代。这里的 Button 可以是以下任意一种:Back、Eject、Next、Pause、Play、Prev、Record、Step 或 Stop。如:

```
MMControl1. PlayEnabled = True          '启用 Play 按钮
MMControl1. EjectEnabled = False        '禁用 Eject 按钮
```

(3) ButtonVisible 属性

设置指定的按钮是否在控件中显示。该属性可以被 Visible 属性所替代。此属性中的 Button 与 ButtonEnabled 属性中的 Button 意义相同。

(4) Command 属性

指定将要执行的 MCI 命令。在设计时,该属性不可用。它的语法为:

```
MMControl1. Command[ = cmdString$]
```

表 9-9 给出了将要执行的 MCI 命令的名称:

表 9-9 常用的 MCI 命令

命令名	描 述	命令名	描 述
Open	打开多媒体设备	Next	定位到下一个曲目的开始部分
Close	关闭多媒体设备	Seek	如果没有进行播放,就搜索一个位置;如果播放正在进行,就从给定位置开始继续播放
Play	播放打开的多媒体设备	Record	开始进行记录
Pause	暂停播放或记录	Eject	将媒体弹出
Stop	停止播放或记录	Sound	播放声音
Back	向后单步播放	Save	保存打开的文件
Step	向前单步播放		
Prev	定位到当前曲目的开始部分		

(5) DeviceType 属性

指定要打开的 MCI 设备的类型。语法格式为:

```
MMControl1. DeviceType[ = device$]
```

其中,device$ 是一个字符串值,代表一个有效的设备类型。表 9-10 列出了由 MMControl 控件支持的常用多媒体设备和在 DeviceType 属性中用到的 device$ 参数。系统是否支持这些设备类型,可查看位于 Windows 的 System. ini 文件的[mci]节中的内容,它是系统安装和维护 MCI 设备时写入的。

表 9-10 常用的多媒体设备类型

媒体设备	device$	描 述
视频(. avi 文件)	AVIVideo	Microsoft AVI 格式视频
音频 CDs	CDAudio	用所连接的 CD-ROM 驱动器播放音乐
CD 数字磁带	DAT	使用所连接的数字磁带(DAT)
MIDI 音序器	Sequencer	MIDI 音序器资料
音频(. wav)文件	WaveAudio	Microsoft Windows 音频文件

通常,可在 Form_Load 事件过程中放入必要的程序代码就可以实现在程序启动时,系统自动播放指定的设备。例如,以下代码能实现在程序启动时播放文件 Start. wav。

```
Private Sub Form_Load ( )
    MMControl1. AutoEnable  = True
    MMControl1. DeviceType  = "WaveAudio"
    MMControl1. FileName  = " C:\WINDOWS\MEDIA\Start. wav"
    MMControl1. Command  = "Open"      '打开 MCI WaveAudio 设备
End Sub
```

(6) FileName 属性

指定用 Open 命令将要打开的或者用 Save 命令将要保存的文件。如果在运行时要改变 FileName 属性,就必须先关闭然后再重新打开 MMControl 控件。用法见上面的相关代码。

(7) hWndDisplay 属性

为利用窗口显示输出结果的输出设备,指定显示输出的窗口。该属性设计时不可用。

(8) Mode 属性

返回打开的 MCI 设备的当前状态。在设计时,该属性不可用;在运行时,它是只读的。表 9-11列出了 Mode 属性返回的值及其含义。

表 9-11　Mode 值及其含义

值	设置值/设备模式	描　述
524	mciModeNotOpen	设备没有打开
525	mciModeStop	设备停止
526	mciModePlay	设备正在播放
527	mciModeRecord	设备正在记录
528	mciModeSeek	设备正在搜索
529	mciModePause	设备暂停
530	mciModeReady	设备准备好

(9) Notify 属性

指定下一条 MCI 命令是否使用 MCI 通知服务。如果它被设置为 True,那么 Notify 属性在下一条 MCI 命令完成时产生一个回调事件(Done)。在设计时,该属性不可用。赋给该属性的值只对下一条 MCI 命令有效。后面的 MCI 命令会一直忽略 Notify 属性,除非赋给它另外一个值(不同的或可标识的)。以下代码段演示了该属性的使用方法。

```
Private Sub Form_Load ( )                '设置必需的属性以便打开 MCI 设备
    MMControl1. Notify  = True           '使用 MCI 通知服务
    MMControl1. Wait  = True             '控件等待 Open 命令执行完毕
    MMControl1. Shareable = False        '系统中的其他应用程序不能访问已打开的 MCI 设备
    MMControl1. DeviceType  = "WaveAudio"
    MMControl1. FileName  = "C:\WINDOWS\MIDA\GONG. WAV"
    MMControl1. Command  = "Open"        '打开 MCI WaveAudio 设备
End Sub
```

为了正确管理多媒体资源,在退出应用程序之前,应该关闭那些已经打开的 MCI 设备。可在 Form_Unload 过程中写入关闭命令,则在退出包含 MMControl 控件的窗体之前,就可以关闭那些已经打开的 MCI 设备。示例代码如下:

```
Private Sub Form_Unload (Cancel As Integer)
    MMControl1. Command = "Close"
End Sub
```

(10) UpdateInterval 属性

指定两次连续的 StatusUpdate 事件之间的毫秒数。语法格式为:

```
MMControl. UpdateInterval[ = milliseconds% ]
```

其中,参数 milliseconds% 规定事件之间的毫秒数。如果毫秒数是 0,就表明没有 StatusUpdate 事件发生。

(11) Wait 属性

决定 MMControl 控件是否要等到下一条 MCI 命令完成,才能将控件返回应用程序。设置为 True,则 MMControl 必须等到下一个 MCI 命令完成才能将控件返回应用程序。在设计时,该属性不可用。

2. MMControl 控件的主要事件

(1) ButtonClick 事件

当用户在 MMControl 控件的按钮上按下并释放鼠标按钮时发生。其中 Button 含义如前述。

(2) StatusUpdate 事件

按 UpdateInterval 属性所给定的时间间隔自动地发生。这一事件允许应用程序更新显示,以通知用户当前 MCI 设备的状态,如播放进度等。应用程序可以从 Position、Length 和 Mode 等属性中获得状态信息(见例 9-7)。

9.2.4 多媒体程序举例

【例 9-6】 设计一个 AVI 文件播放器。

① 新建一工程(名为 vb9_6. vbp),命名为"AVIPlayer",内建一名为"frmExam_9_6"的窗体。

② 在窗体上(图 9-10)加入控件,控件及主要属性设置如表 9-12 所示。未指明的均为默认设置。



图 9-10 AVI 文件播放器窗体布局

表 9-12　控件属性的设置

控 件 名	属 性 名	设 置 值
PictureBoxl	Name	Picture1
CommonDialog	Name	CommDialog1
	DialogTitle	打开 AVI 文件
	FileName	*.avi
	Filter	*.avi

③ 例子中的代码清单如下：

```
Private Sub Form_Load( )        '窗体的 Load 事件,初始化 MMControl 控件
    MMControl1. Notify = False
    MMControl1. Wait = True
    MMControl1. Shareable = False
    MMControl1. DeviceType = "AVIVideo"
End Sub
```

```
Private Sub cmdLoad_Click( )
    '运行打开文件对话框,选择 AVI 文件
    CommonDialog1. ShowOpen
    MMControl1. FileName = _
          CommonDialog1. FileName
    '打开设备
    MMControl1. Command = "Open"
    '指定输出窗口为图像框 PictureBox
    MMControl1. hWndDisplay = Picture1. hWnd
End Sub
```

```
Private Sub cmdClose_Click( )
    End
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
  MMControl1. Command = "Close"    '关闭窗口时释放资源
End Sub
```

【例 9-7】 完成案例十中模拟 CD 播放器的设计。

设计过程如下：

① 新建一工程(名为 vb9_7. vbp)，在该工程中建立一个名为“frmExam_9_7. frm”的窗体。

② 在窗体上(图 9-8)加入控件，窗体、控件及属性设置如表 9-13 所示。未指明的均为默认设置。

表 9-13 窗体及控件属性的设置

控 件 名	属 性 名	设 置 值
Label	Name	lblIndicator
	Caption	播放状态
	Alignment	Centre
	BorderStyle	1 - Fixed single
	BackColor	&H00000000&
	ForeColor	&H0080C0FF&
	Font	宋体,小四
PictureBox	Name	picCD
	Picture	Audio. bmp
	Visible	False
MMControl	Name	mciCDPlayer

③ 在 MMControl 控件的 StatusUpdate 事件中,取当前曲目号和已播放的时间,并显示在播放指示器标签 lblIndicator 中。

④ 例程中的代码清单如下:

```
Private Sub cmdLoad_Click( )
    '打开 CD 设备(光盘已经在驱动器中)
    mciCDPlayer. Command = "Open"
    '设置时间格式
    mciCDPlayer. TimeFormat = vbMCIFormatTmsf
    '禁用"加载"按钮
    cmdLoad. Enabled = False
    picCd. Visible = True
End Sub
```

```
Private Sub Form_Load( )
    mciCDPlayer. Wait = True
    mciCDPlayer. UpdateInterval = 0
    mciCDPlayer. DeviceType = "CDAudio"          '设置 DeviceType 属性为音乐 CD 设备
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
    mciCDPlayer. Command = "Close"               '窗体卸载时释放资源
End Sub
```

```
Private Sub mciCDPlayer_EjectClick(Cancel As Integer)   '弹出事件
    cmdLoad. Enabled = True                      '使"加载"按钮可用
    mciCDPlayer. UpdateInterval = 0
    mciCDPlayer. Command = "Eject"               '从 CD 设备中退出光盘,并且关闭此设备
    mciCDPlayer. Command = "Close"

    lblIndicator. Caption = "弹出"               '设置"播放指示器"
```

```
    picCd. Visible = False
End Sub

Private Sub mciCDPlayer_PauseClick(Cancel As Integer)        '暂停事件
    mciCDPlayer. UpdateInterval = 0
    lblIndicator. Caption = "暂停"                           '设置"播放指示器"
End Sub

Private Sub mciCDPlayer_PlayClick(Cancel As Integer)         '播放事件
    mciCDPlayer. UpdateInterval = 1000                       '设置 StatusUpdate 事件发生的时间间隔
End Sub

Private Sub mciCDPlayer_StatusUpdate( )
    Dim length As Long
    Dim StartPosition As Long
    Dim PresentPosition As Long
    Dim min As Integer
    Dim sec As Integer
    Dim strTrack As String

    strTrack = Format$ (mciCDPlayer. Track, "00")            '取当前曲目号
    StartPosition = mciCDPlayer. TrackPosition               '取当前曲目起始位置(毫秒)
    StartPosition = mciCDPlayer. TrackPosition / 1000
    PresentPosition = mciCDPlayer. Position                  '取当前曲目当前位置(毫秒)
    PresentPosition = mciCDPlayer. Position / 1000
    length = PresentPosition - StartPosition                 '计算当前曲目已播放时间
    min = length\60                                          '转换为分,秒
    sec = length Mod 60

    '设备当前如正在播放则设置音轨号及已播放时间
    If mciCDPlayer. Mode = mciModePlay Then
       lblIndicator. Caption = "[曲目" & strTrack & "] "  &  _
            Format$ (min, "00") & ":" & Format$ (sec, "00")
    End If
End Sub

Private Sub mciCDPlayer_StopClick(Cancel As Integer)         '停止播放事件
    mciCDPlayer. UpdateInterval = 0                          '重新设置 CD 到音轨 1—曲目 1
    mciCDPlayer. Command = "Stop"
    mciCDPlayer. To = mciCDPlayer. Start
    mciCDPlayer. Command = "Seek"
    mciCDPlayer. Track = 1
    lblIndicator. Caption = "停止"
End Sub
```

习 题 9

1. 完成本章的 7 个例子。

2. MMControl 控件有哪些主要的属性和事件？简述使用该控件的步骤。

3. 修改学生成绩管理系统主窗体 frmMainForm，使之在窗体打开后（反复）自动播放一段音乐直到用户单击“注册”菜单项为止，且播放时 MMControl 控件不可见。

4. 设计一个程序，根据打开的多媒体文件类型使用 MMControl 控件播放该文件。

5. 编写一个程序，能在窗体上画一条帆船，并使它沿直线运动。

6. 如何把滚动条和图片框控件组合起来完成图片大小的调整？

7. 编写一段程序，能利用 Image 控件来实现图像大小的控制。

8. 什么是 ActiveX 控件？ActiveX 控件的主要优点是什么？ActiveX 控件从功能角度分为哪几类？

9. 完成绘制圆弧的任务（见图 9-11）。

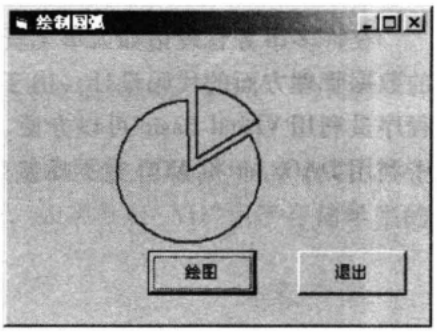


图 9-11 绘制圆弧

10. 在图片框中绘制函数图形（图 9-12）。

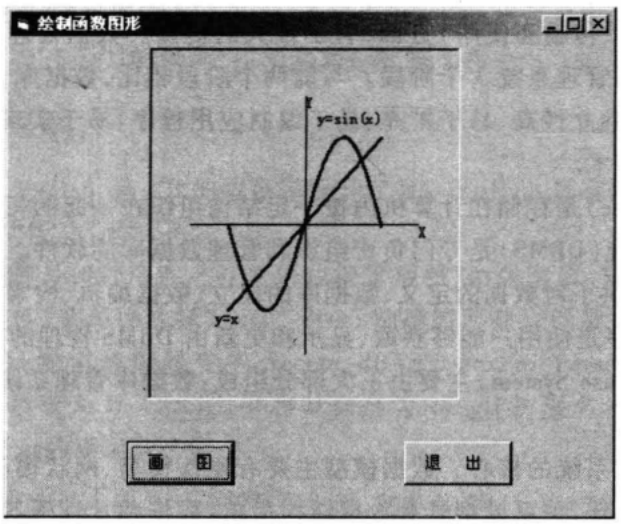


图 9-12 绘制函数图形

* 第 10 章　数据库应用程序的开发

了解有关数据库的基础知识；掌握利用 Data 控件和 Adodc 控件访问数据库的方法，能进行一般的数据库应用程序的设计。

在许多事务管理诸如人事工资管理、学生成绩管理等应用系统设计中，程序员要进行大量的数据管理方面的代码设计。由于 Visual Basic 结合了传统的编程语言和数据库语言的优点，程序员利用 Visual Basic 可以方便、高效地设计数据库应用系统。本章将介绍在 Visual Basic 6.0 中利用 DAO/Jet 和 ADO 数据库接口技术访问数据库的方法及程序设计。

10.1　数据库基础

10.1.1　数据库的概念

随着计算机硬件和软件技术的飞速发展，在对反映客观世界的各种数据的处理技术（包括数据的收集、处理、存储、传输和使用）方面，有了很大的提高。数据处理技术经历了人工管理、文件系统管理和数据库管理系统 3 个阶段。与前两个阶段相比，数据库管理系统阶段具有数据冗余度低、程序和数据独立性高、易于扩充、易于编制应用程序、易于实现资源共享等特点，是当今最先进的数据处理技术。

① 数据库（Database）是存储在计算机内按一定结构组织在一起的相关数据的集合。

② 数据库管理系统（DBMS）是专门负责组织和管理数据库的软件。DBMS 是用户与数据库之间的接口，为用户提供了对数据的定义、数据库的建立、数据编辑、检索等一系列操作。

③ 数据库应用程序是使用户能够获取、显示和更新由 DBMS 管理的数据的用户程序。

数据库系统（Database System）主要由三大部分组成：数据库管理系统、数据库应用程序和数据库。

数据模型是数据库系统的核心。数据模型主要有层次模型、网状模型和关系模型 3 种。与层次模型和网状模型相比，关系模型具有数据结构简单、数据独立性强等优点，现今流行的大多数数据库系统都是关系型的。如 Visual FoxPro、Access、SQL Server、Sybase 和 Oracle 等。

DBMS 和数据库应用程序可存储在同一台计算机上，并能在同一台计算机上运行，或分别存储在不同的计算机上运行。后者的 DBMS 称为客户/服务器模式，数据库应用程序运行在一个或多个用户工作站（客户机）上，并且通过网络与运行在其他计算机（服务器）上的一个 DBMS 进行通信。客户/服务器模式提高了数据库系统的数据处理能力。像 Sybase、Microsoft SQL Server 2000、Oracle 9i 就是客户/服务器模式的 DBMS。

10.1.2 关系型数据库基础

关系型数据库是通过满足一定条件的二维表格来表示数据间联系的一种数据模型。每个二维表格都表示一种关系,这个二维表格称为表(Table),表是关系型数据库的基本元素。设有一表名为 XSMD 的学生名单表,如表 10-1 所示。

表 10-1 学生名单表

学 号	姓 名	性 别	班 级	联 系 地 址	邮 政 编 码
2001010101	王 芳	女	计算机应用 2001	无锡市人民路 4-1 号	214002
2001010102	苏小明	女	计算机应用 2001	武进市新市路 14 号	213101
2001020101	孙明亮	男	电子工程 2001	南京市新海路 110 号	210024

- 字段(Field):表中的每列称为字段。每个字段都有相应的描述信息,包括字段名、存储的数据类型(有整型、字符型、布尔型、日期型等)、数据宽度等。如字段名"XH"用于存储学生的学号,数据类型是字符型,宽度为 10(最多存 10 个字符)。
- 记录(Record):表中的每一行称为记录(Record)。
- 索引(Index):使用索引可以提高数据检索的速度。当检索数据时,DBMS 先从索引文件中找到信息的位置,再从表中读取数据。每个表可有多个索引,每个索引是一个或多个字段的组合,每个索引用不同的名字进行标识。
- 主关键字:用来唯一地标识表中记录的一个或一组字段。主关键字不允许重复,如可用学生的"学号"作为主关键字,而不可以用"姓名"作为主关键字,因为在表中可能会出现同名的学生。

10.1.3 建立数据库

可视化数据管理器 VisData 是 VB 的数据库设计工具。VisData 具有创建和维护数据库和数据表、执行 SQL 语句等功能。

在 VB 主菜单中选择〖外接程序〗→〖可视化数据管理器〗命令菜单,即可启动可视化数据管理器。在 VisData 的"文件"菜单中选择"新建"命令,选择需要建立的数据库的类型为 Microsoft Access 的版本 7.0 MDB(数据管理器支持 Access、dBASE、FoxPro、Paradox 以及 ODBC 等数据库);选择数据库的存放位置为 D:\Xscjgl,并将其命名为 xscjgl(扩展名将自动取为 .mdb)。此时,将显示如图 10-1 所示的界面。

☞ 建立和管理 Access 数据库用 Access 97 更方便,但不能用 Access 2000,因为它与 VB 不兼容;但 Access 2000 提供了将数据库转为 Access 97 格式的功能。

10.1.4 建立并维护数据库表

在图 10-1 中有两个窗口:数据库窗口用来显示数据库的结构,如数据库的属性,数据库的表、字段、索引和查询以及相应的属性;SQL 语句窗口用来输入 SQL 语句,对数据库进行数据表数据的查询等操作。

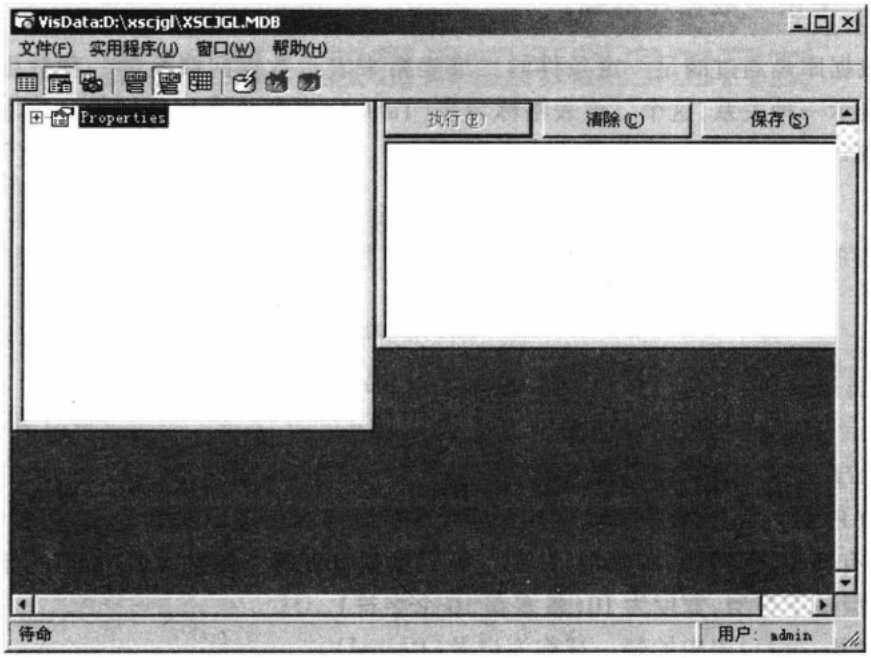


图 10-1　可视化数据管理器

1. 建立表结构

在数据库窗口中单击鼠标右键,在弹出菜单中选择“新建表”命令,进入建立“表结构”对话框,如图 10-2 所示。现在按表 10-2 的要求建立一个表名为 XSMD 的学生档案的表结构。

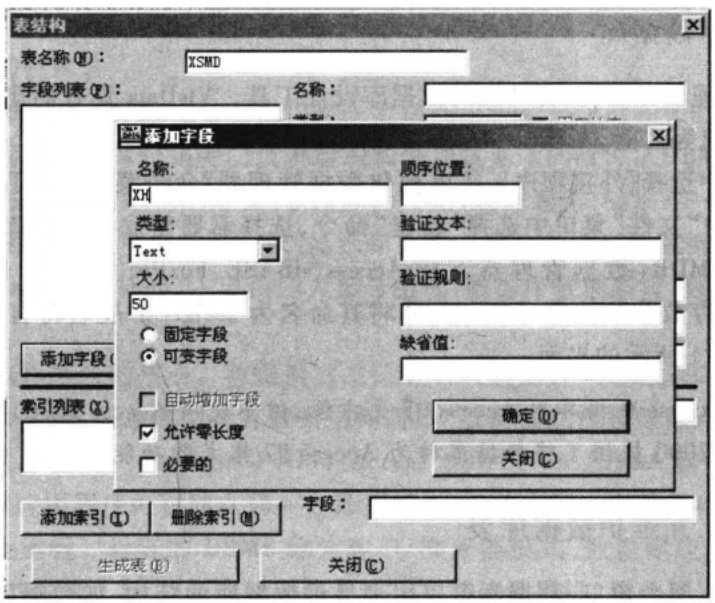


图 10-2　“表结构”和“添加字段”对话框

表 10-2 学生档案表结构

序号	字段名	中文含义	数据类型	宽度	验证规则
1	XH	学号	Text(字符型)	10	Like "##########"
2	XM	姓名	Text(字符型)	8	
3	XB	性别	Boolean(逻辑型)	1	
4	SFTY	是否团员	Boolean(逻辑型)	1	
5	CSRQ	出生日期	Date(日期型)		
6	BJDM	所在班级	Text(字符型)	8	
7	RXSJ	入学时间	Text(字符型)	7	
8	LXDZ	联系地址	Text(字符型)	40	
9	YZBM	邮政编码	Text(字符型)	6	

在该对话框的表名称中输入表名 XSMD,单击“添加字段”命令按钮,进入“添加字段”对话框。逐个按需要输入相应的字段名,确定其数据类型、大小、长度是否固定、是否必要(即是否允许为空)、顺序位置(各字段的相对位置)、有效性验证规则(即输入数据的合法性验证规则,常用 Like"字符串"形式表示,如 Like"####"表示输入的必须是 4 位数字)和有效性验证文本(数据不符合规则时,应用程序系统给予的提示信息)以及缺省值等,然后单击“确定”按钮完成该字段的建立,重复这个过程直到表中所有字段均建立完毕。字段添加结束,单击“关闭”按钮还回到“表结构”对话框,字段列表框中将显示出该表中已经建立的所有字段,右边的文本框则显示出当前选中字段的所有属性。

单击“添加索引”命令按钮,以字段 XH 建立索引名称为 XSMD_KEY 索引,并设置为“主要的、唯一的”,即 XH 是主关键字。

可以使用“删除字段”命令按钮将添加的某个字段删去。若要更改某个字段,应先将其删除然后再添加。表建立后若要对该表进行维护,可以在数据库窗口中选中该表,单击鼠标右键,在弹出菜单中选择“设计”命令从而对该表结构进行维护。

2. 表数据维护

为了便于维护表中的数据,应先指定数据的显示方式和记录集(RecordSet)的类型。

数据显示方式有 3 种:使用数据控件显示数据、使用 DBGrid 控件(数据网格也称电子表格)显示数据和不使用数据控件显示数据。

记录集类型有下列 3 种。

- DynaSets:动态集类型记录集,常用 SQL 语句从一个或多个表中提取符合条件的记录而生成,是可动态修改的数据,是最灵活、强大的记录集。
- SnapShots:快照类型记录集,静态(只读)地显示数据,是同一数据的静态视图,不能更新记录。
- Tables:表类型记录集,数据库表数据的直接显示,是数据表的直接引用,只限于单个表。

单击图 10-2 所示工具栏上的相应按钮(图 10-3)就可以选用记录集类型和显示数据的方式。然后双击数据库窗口中的表名或在右击出现的弹出菜单中选择“打开”命令,此时弹出相应的数据显示窗口供用户进行数据维护。

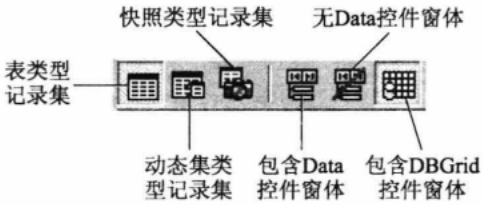


图 10-3　VisData 的工具栏

图 10-4 是选择动态集类型记录集、用 Data 控件情况下的数据表记录维护窗口。图中的“更新”按钮用于保存修改结果,“添加”按钮用于添加记录。可以通过使用 Data 控件上相应的导航按钮定位记录,或通过“查找”按钮,输入要查找的搜索表达式进行记录定位。

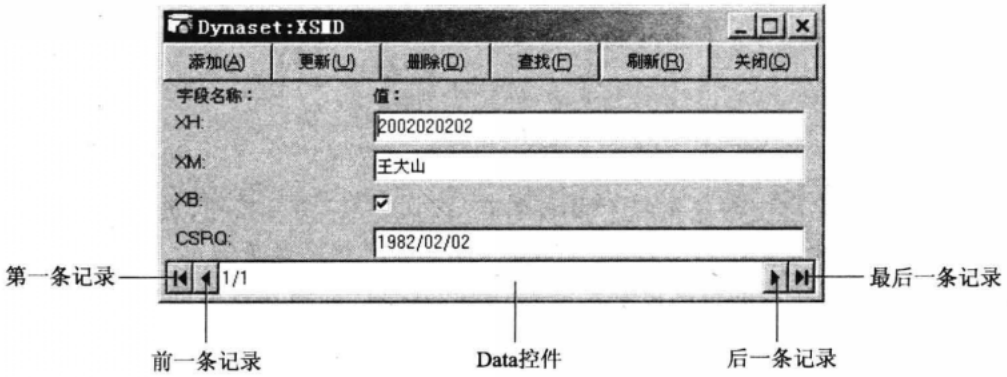


图 10-4　数据表记录维护

3. 建立本章例题中所需的表

运用 VB 可视化数据管理器 VisData 再在 D:\Xscjgl 目录下名为 xscjgl. mdb 数据库中建立本章例子中所需表的结构如表 10-3 所示。

表 10-3　本章案例中所需的表结构

(1) USER——用户名表

序　号	字 段 名	中文含义	类　型	宽度	索引	引用字段	取　值
1	User_ID	用户标识	自动				
2	User_Name	用户名	Text	20	UserName		
3	Password	口令	Text	15			
4	Right	用户权限	Boolean	1			1 - True 管理员

(2) BJMC——班级编码表

序　号	字 段 名	中文含义	类　型	宽度	索引	引用字段	取　值
1	BJDM	班级代码	Text	8	BJMC_KEY		
2	BJMC	班级名称	Text	20			
3	ZYDM	专业代码	Text	4		ZYML. ZYDM	
4	RXSJ	入学时间	Text	7			年/月

续表

(3) ZYML——专业目录表

序　号	字 段 名	中 文 含 义	类　型	宽 度 索 引	引 用 字 段
1	ZYDM	专业代码	Text	4	
2	ZYMC	专业名称	Text	20	

在 USER 表中输入如图 10-5 中的记录;在 BJMC 表中输入如图 10-6 中的记录;在 ZYML 表中输入如图 10-7 中的记录。

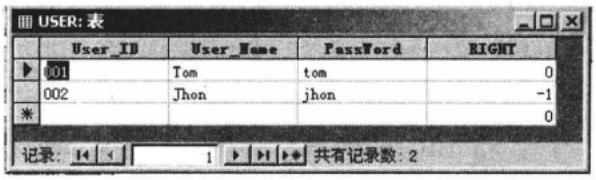


图 10-5 USER 表的记录



图 10-6 BJMC 表的记录



图 10-7 ZYML 表的记录

10.2 用数据窗体向导设计数据库应用程序

当在数据库中建立完表以后,可以使用 Data 控件或 ADO 控件访问数据库表中的数据,从而设计出数据库应用程序的窗体。VB 的“数据窗体向导”可以快速帮助用户自动生成数据库应用程序窗体。

【例 10-1】 使用“数据窗体向导”快速自动生成“用户管理”数据库应用程序窗体,该窗体可以对 USER 表进行维护。

① 初次使用 VB,“外接程序”菜单中没有“数据窗体向导”菜单项。可以通过选择〖外接程序〗→〖外接程序管理器〗命令菜单启动和加载“数据窗体向导”菜单项(在图 10-8 中选择“VB 6 数据窗体向导”,然后将“加载行为”框架中的前两个复选框选中)。

加载“数据窗体向导”菜单项后,就可通过选择〖外接程序〗→〖数据窗体向导〗命令菜单启动“数据窗体向导”了,进入“数据窗体向导 - 介绍”对话框。

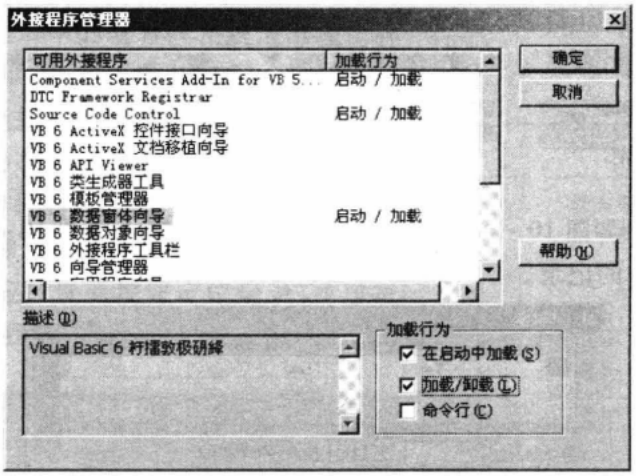


图 10-8　启动和加载“数据窗体向导”菜单

② 单击“下一步”，进入“数据窗体向导 - 数据库类型”对话框，选择“Access”。

③ 单击“下一步”，进入“数据窗体向导 - 数据库”对话框，输入或通过单击“浏览”按钮，选择一个数据库，如“D:\Xscjgl\xscjgl. mdb”。

④ 单击“下一步”，进入“数据窗体向导 - Form”对话框，输入窗体名称“frmUser_10_1”，窗体布局为“网格(数据表)”，绑定类型为“ADO 数据控件”。

⑤ 单击“下一步”，进入“数据窗体向导 - 记录源”对话框，在“记录源”下拉列表框中选择表“USER”，在“可用字段”列表框中依次选择“User_Name”、“PassWord”、“RIGHT”，如图 10-9 所示。

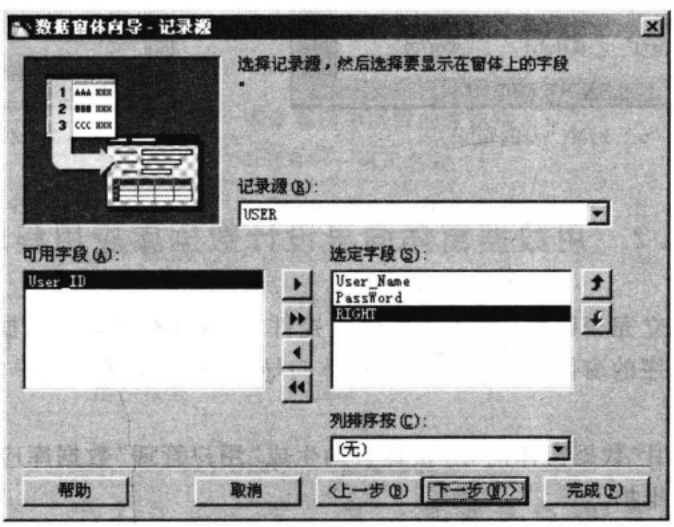


图 10-9　数据窗体向导 - 记录源

⑥ 单击“下一步”，进入“数据窗体向导 - 控件选择”对话框，在该对话框中选择需要的控件，如记录添加、删除等。

⑦ 单击“完成”，系统自动生成名为“frmUser_10_1. frm”的窗体，并自动为命令按钮添加了

相关的事件代码。

⑧ 将 rmUser_9_1. frm 窗体保存在 VB10_1. VBP 工程文件中(本章的示例的窗体及工程文件等若不特别说明均存放在 D:\Xscjgl 目录下)。

运行该程序,其结果如图 10-10 所示,可以单击相应的命令按钮完成对 USER 表的维护工作。

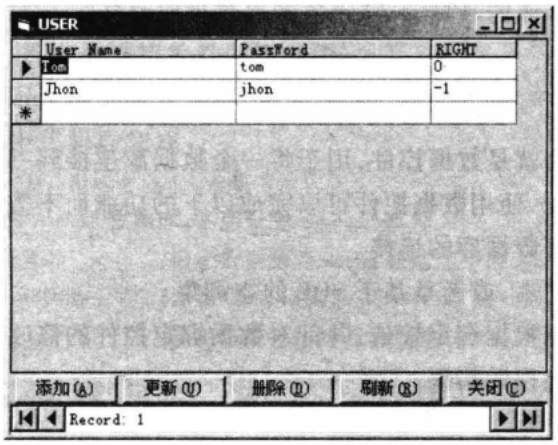


图 10-10 用户管理

10.3 用 Data 控件访问数据库

当在数据库中建立完表以后,用户可以使用"数据窗体向导"快速地自动生成数据库应用程序窗体,但要设计出自己风格的应用程序窗体,使用 Data 控件不失为一种好的方法。为了访问数据库,需先通过 Data 控件建立与数据库中数据表的连接,再通过数据绑定控件访问表中的记录。

【案例十一】 使用 Data 控件设计库表 BJMC 编辑维护应用程序窗体(图 10-11)。其中,

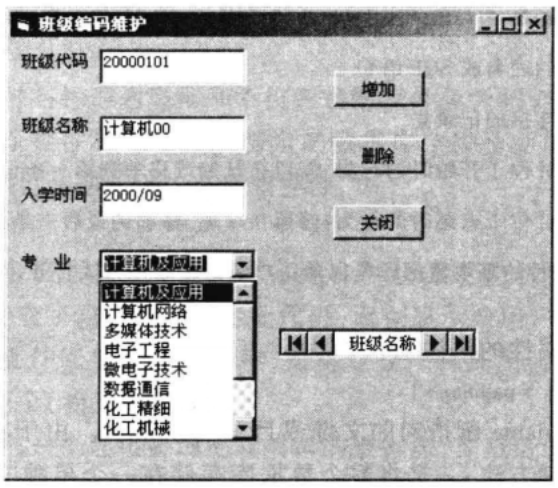


图 10-11 库表 BJMC 编辑维护应用程序窗体

班级编码表 BJMC 和专业目录表 ZYML 的结构如表 10-3 所示，ZYML 的记录如图 10-7 所示（需事先输入），因为库表 BJMC 编辑维护应用程序窗体中的 ZYDM 来源于 ZYML 表。

选择 ZYML 表中 ZYMC 字段的值，将选定的 ZYMC 字段值对应的 ZYDM 字段值传给 BJMC 表的 ZYDM 字段值，这样能够保证表间数据的一致性。

10.3.1　Data 控件

工具箱中的 Data 控件就是数据控件，用于将一个数据源连接到一个数据绑定控件。数据源可以是 DAO/Jet 或 ODBC。使用数据控件可以完成以下的功能而不需编写任何代码。

① 完成对本地和远程数据库的连接。

② 打开指定数据库的表，或者是基于 SQL 的查询集。

③ 将表中的字段传至数据绑定控件，并针对数据绑定控件的修改更新数据库。

④ 捕获访问数据时出现的错误。

⑤ 关闭数据库。

Data 控件在工具箱中的图标为，它置于窗体时的外形为。

1. Data 控件的属性

Data 控件中常规属性如 Name、Caption、Visible 等与其他控件一样，与数据库访问密切相关的主要属性如表 10-4 所示。

表 10-4　Data 控件的主要属性

属 性 名	设置值及意义
Connect	指定所连接的数据库类型
DatabaseName	设置要打开的数据库的名字
ReadOnly	设置数据库中的数据可否修改。设置为 True，则为只读状态
RecordsetType	设置 RecordsetType 对象的类型，有 3 种类型：0 - Table，1 - Dynaset，2 - Snapshot
RecordSource	指定所使用的表名或 SQL 语句
RecordSet	返回控件所连接的记录集
BOFAction	决定记录指针位于开始时的行为：停留在开始或移动到第一条记录
EOFAction	决定记录指针位于末尾时的行为：停留在末尾、移动到最后一条记录或者添加一条新的记录
Exclusive	指定 Data 控件的基本数据库是以单用户还是多用户方式打开，True 表示为单用户

其中，DatabaseName 属性的语法为：

object. DatabaseName[= pathname]

数据库类型可由 pathname 所指向的文件或目录加以指示。由于在微机上的关系型数据库中，有两种不同的数据存储方法：一是将整个数据库存储在一个单独文件中，如 Microsoft Access（*. mdb 文件），pathname 就是这个文件名（含路径）；二是分别将数据表格、索引以及数据库的

其他元素存储在同一个目录下的各自文件中，如 dBASE、FoxPro、Paradox 等数据库，pathname 则为该目录名。

2. Data 控件的主要方法

- Refresh 方法：打开或重新打开数据库（如果 DatabaseName、ReadOnly、Exclusive 或 Connect 属性的设置值发生改变），并能重建控件的 Recordset 属性内的 Dynaset。
- UpdateRecord 方法：保存被连接控件的当前值。

例如，要求打开名为 xscjgl. mdb 的数据库，记录源为“XSMD”表，对此记录集希望只能添加记录而不能读取，可以用以下代码段完成：

```
Data1. DatabaseName = "D:\Xscjgl\xscjgl. mdb "
'打开名为 xscjgl. mdb 的 Access 数据库，不用设置 Connect 属性
Data1. DataSource = "XSMD"          '指定数据源为 xscjgl 数据库中的 XSMD 表
Data1. Options = dbAppendOnly       '设置 Options 属性 dbAppendOnly 以满足对记录集的限制
Data1. Refresh                      '使用刷新方法打开数据库
```

10.3.2 数据访问控件

利用 Data 控件建立与数据库的连接后，要访问数据库中的数据必须通过数据访问控件来实现。VB 中的数据访问控件包括标签、文本框、复选框、图片、图像、OLE 容器、列表框和下拉列表框等控件，可把它们和由数据控件管理的数据库记录集的某一字段绑定。所谓数据绑定，即控件中的数据显示和操作的结果与数据库保持实时一致性。当改变控件中的数据内容时，数据库对应的数据也会发生同样的变化。

此外，VB 还提供另外的数据绑定控件供程序员使用，可与 Data 控件一起使用的这些控件是：DBList 和 DBCombo，它们都能管理记录集合。要使用这些控件，需将它们添加到工具箱中，在菜单中选择〖工程〗→〖部件...〗命令，在“部件”对话框中选中“Micrsoft Data Bound List Controls 6.0”即可。

上述控件都有 DataSource 属性和 DataField 属性，使用时要设置 DataSource 属性为 Data 控件的 Name 值，设置 DataField 属性为对应 Data 控件中已指定数据集的某个字段名。下面主要介绍 DBList 和 DBCombo 控件。

DBCombo 和 DBList 控件与标准列表框和下拉列表框控件极为相似，但功能的侧重点不同。DBCombo 和 DBList 控件两个控件都可以被所绑定的数据库字段自动填充。此外，它们还能有选择地将一个选定字段的数据传递给第二个数据控件。

（1）控件功能

在学生成绩数据库中，各班所属专业的名称存储在专业目录表中，每个专业都有一个唯一的代码。班级编码表中的专业字段存储的是专业代码，而在显示给用户时则应该用专业名称来显示。可以使用 DBCombo 控件来显示专业名称，而将专业代码提供给班级编码表。从而实现用一个表的数据输入给另一个（相关的）表。

这时需要两个数据控件，一个连接要浏览或编辑的数据源，如班级编码表；另一个连接项目数据源，如专业目录表。后述的 ADO 对象中的 DataList 和 DataCombo 控件的用法与这两个控件用法相同。

(2) 主要属性

DBList 和 DBCombo 控件的一些主要属性如表 10-5 所示。

表 10-5 DBList 和 DBCombo 控件的主要属性

属 性 名	设置值及意义
DataSource	要浏览或编辑数据源的 Data 控件名,即与 DBList 或 DBCombo 绑定的数据控件
DataField	由 DataSource 属性指定的记录集中的一个字段,用该字段确定列表显示的元素
RowSource	列表字段作为项目数据源使用的 Data 控件名
ListField	由 RowSource 指定的记录集中用于填充下拉列表的字段名,为用户可见的
BoundColumn	由 RowSource 指定的项目源记录集中的字段名,当列表项目选择确定后,该字段更新 DataField
BoundText	BoundColumn 字段对应的文本值

10.3.3 使用 Recordset 对象访问数据库

Data 控件的优点是用户无需编写大量代码就可开发一个简单的数据访问应用程序,而不必提供程序代码来打开或创建数据库或记录集、在记录间移动或编辑记录集中已有的记录。尽管如此,它仍有一些局限性,如在 Data 控件内没有内置的添加或删除功能等。

为此,需要将控件和程序代码结合起来,而 Data 控件的 Recordset 属性为编制代码提供了丰富的属性和方法,是控件代码编程的核心。

Recordset 属性是由 Data 控件返回的代表选定记录集的一个集合对象。使用 Recordset 属性,用户可以把 Data 控件的记录集传递给其他过程或模块,同样也可以用 DAO 创建记录集,然后把记录集赋给 Data 控件。这为用户使用记录集提供了极大的灵活性。

1. Recordset 对象的主要属性

Recordset 对象的主要属性如表 10-6 所示。

表 10-6 Recordset 对象的主要属性

属 性 名	设置值及意义
EOF	表示该记录指针是否在 Recordset 中最后一个记录之后。True 表示“是”
BOF	表示该记录指针是否在 Recordset 中首记录之前。True 表示“是”
NoMatch	该属性的值为 True 时表示未查找到匹配项
RecordCount	返回一个表示该 Recordset 对象所包含的记录数
AbsolutePosition	返回或设置一个表示该 Recordset 对象的当前记录的绝对位置
Fields	包含一个 Recordset 所存储的字段对象集合
Sort	返回或设置一个 Recordset 对象中的记录排列顺序
Type	返回表示该 Recordset 属于表、Dynaset 或 Snapshot 类型的值

Fields 属性是从 0 开始的 Field 对象集合,代表使用普通数据类型的数据列。使用 Name 属性返回字段名,用 Value 属性(也是缺省属性)查看或更改当前记录字段中的数据,用 Count 属性返回字段数。字段数据的访问方法有 3 种:

- Recordset(index)
- Recordset. Fields(index)
- Recordset. Fields(fieldname)

其中,fieldname 为要访问字段的字段名,相应地获取字段名的方法有两种:

- Recordset(index). Name
- Recordset. Fields(index). Name

index 为字段在记录集中从 0 开始的顺序号。一条记录的字段数为 Recordset. Fields. Count。

例如,访问当前记录中的所有字段可用下列代码段:

```
Dim  fldLoop  As  Field                              '声明 Field 对象变量
For Each fldLoop In datXsmd. Recordset. Fields      '枚举学生档案表当前记录字段集合
    Debug. Print "    " & fldLoop. Name & "  =  " & fldLoop. Value
Next fldLoop
```


由于 Value 是字段对象的默认属性,所以此处关键字 Value 可省略。

上例中用到了 For Each ... Next 循环,它用来对集合对象或数组中的每个元素进行访问(本例是对当前记录每个字段进行访问)。该语句的语法格式为:

```
For Each element In  group
    [statements]
    [Exit For]
    [statements]
Next [element]
```

其中,*group* 为对象集合名或数组名。

2. Recordset 对象的主要方法

Recordset 对象的主要方法如表 10-7 所示。

表 10-7 Recordset 对象的主要方法

方 法 名	功 能
MoveFirst	将控件定位到第一个记录
MoveLast	将控件定位到最后一个记录
MovePrevious	将控件定位到前一个记录
MoveNext	将控件定位到后一个记录
AddNew	增加新记录
Delete	删除当前记录
Update	更新当前记录

续表

方 法 名	功　能
Edit	将记录拷贝到缓冲区,以便编辑该记录
FindFirst	查找指定条件的第一个记录
FindLast	查找指定条件的最后一个记录
FindPrevious	查找指定条件的前一个记录
FindNext	查找指定条件的后一个记录
Move	记录指针移动到指定的行数(+/ -)
Close	关闭该 Recordset 对象记录
Requery	通过再次运行查询来更新一个 Recordset 对象中的数据



当 EOF 和 BOF 均为 True 时,记录集中没有记录,此时不能执行 Delete 和 Move 方法。

其中,Find 方法的用法为:

```
Recordset.{FindFirst| FindLast|FindNext|FindPrevious}  express
```

express 用于描述当前进行查找项目的一个字符串表达式,其格式同 SQL 的 WHERE 子句部分类似,详见 SQL 查询一节。如查找满足条件的第一条记录可用下列语句:

```
Dim Name as String
Name = InputBox("请输入要查找的学生姓名","输入查找条件")
datXsmd.Recordset.FindFirst "XM = '" & Name & "'"
```

【例 10-2】 利用上述知识完成案例十一:班级编码编辑维护应用程序窗体的设计。

本例中,BJMC 表中的 ZYDM 字段数据来自另一个表 ZYML 的 ZYDM 字段,因此可用 DBCombo 控件实现将专业代码用相应的专业名称显示。通过将 Data 控件的 EOFAction 属性设置为"2 - AddNew"来实现表记录的添加。

设计过程如下:

① 新建一工程,命名为"vb10_2.vbp",工程中的窗体命名为"frmbjmc10_2.frm",将它们存放在"D:\Xscjgl"下。

② 按图 10-11 所示界面加入相关控件,主要控件及其属性的设置如表 10-8 所示。

表 10-8　图 10-11 的主要控件及其相关属性的设置

控 件 名	属 性 名	设 置 值
Data	Name	datBJMC
	Caption	班级名称
	Connect	Access
	DatabaseName	xscjgl.mdb
	RecordSource	BJMC
	Visible	True

续表

控 件 名	属 性 名	设 置 值
Data	Name	datZYML
	Connect	Access
	DatabaseName	xscjgl. mdb
	RecordSource	ZYML
	BOFAction	0 – MoveFirst
	EOFAction	2 – AddNew
	Visible	False
DBCombo	DataSource	datBJMC
	DataField	ZYDM
	RowSource	datZYML
	ListField	ZYMC
	BoundColumn	ZYDM
TextBox	DataSoure	datBJMC
	DataField	BJDM
TextBox	DataSoure	datBJMC
	DataField	BJMC
TextBox	DataSoure	datBJMC
	DataField	RXSJ

③ 实现记录的增加、删除功能,需要使用 Recordset 的 AddNew 和 Delete 方法。命令按钮的相应事件代码如下:

```
Private Sub cmdAdd_Click( )
  DatBJMC. Recordset. AddNew          '增加一条新记录
End Sub
```

```
Private Sub cmdClose_Click( )          '关闭窗体
  Unload Me
End Sub
```

```
Private Sub cmdDelete_Click( )
strPrompt$ = "真的要删除当前记录"
vbReply = MsgBox(strPrompt$, vbOKCancel, "删除记录")
If vbReply = vbOK Then
    '如果用户单击确认按钮
    With DatBJMC. Recordset
      If Not (. EOF And . BOF) Then
        '记录集中没有记录时 EOF 和 BOF 均为 True,不可用 Delete 方法,否则出错
        . Delete                      '表中有记录,可删除当前记录
        If . RecordCount > 0 Then     '表中有剩余记录
```

```
            If . EOF Then
                . MoveLast              '记录指针指向末尾,移到最后一条记录
            Else
                . MoveNext              '记录指针移到下一记录
            End If
        End If
            End If
        End With
    End If
End Sub
```

④ 程序的运行结果见图 10-11 所示。

通过上例,可见使用 Data 控件的主要步骤是:

① 添加需要的 Data 控件。

② 设置 Data 控件的属性。

③ 添加其他控件,对需要显示的字段与数据绑定控件绑定。

④ 使用 Data 控件的方法和事件。

⑤ 必要时,可以添加代码以实现更为复杂的功能。

上述步骤的④、⑤将在后面的内容中陆续介绍。

10.4　使用 ADO 控件访问数据库

【案例十二】 使用 ADO 控件设计库表 XSMD 编辑维护应用程序窗体(见图 10-12)。其中学生名单表 XSMD 和班级名称表 BJMC 的结构分别如表 10-2 和表 10-3 所示,BJMC 的记录如

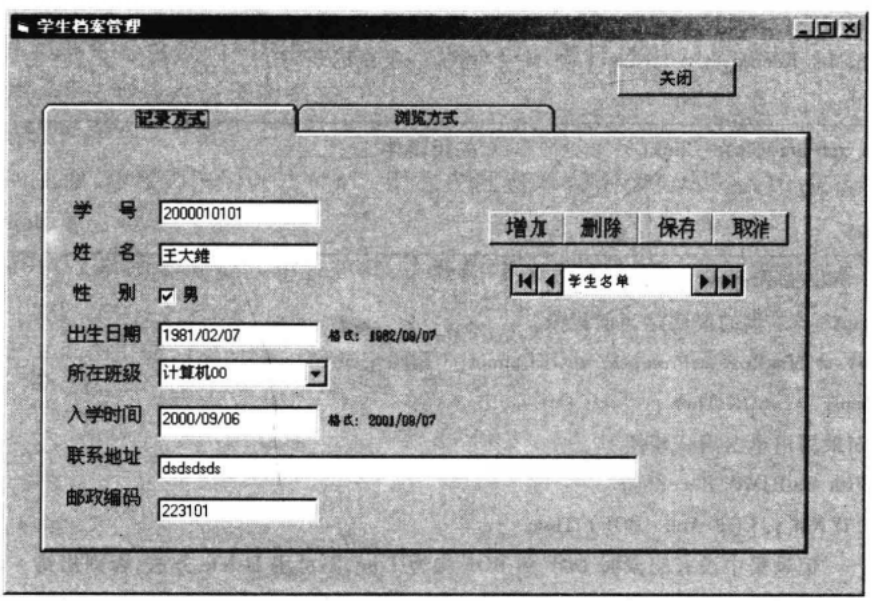


图 10-12　库表 XSMD 编辑维护应用程序窗体

图 10-6 所示(需事先输入),因为库表 XSMD 编辑维护应用程序窗体中的 BJDM 来源于 BJMC 表。

选择 BJMC 表中 BJMC 字段的值,将选定的 BJMC 字段值对应的 BJDM 字段值传给 XSDA 表的 BJDM 字段值,这样能够保证表间数据的一致性。

10.4.1 Visual Basic 数据库访问技术

VB 6.0 提供了多种访问数据库的方案。DAO(Data Access Object,数据访问对象)是本地 Jet(数据库引擎,如 Microsoft Access 数据库)和 ISAM(顺序索引查找方法,如 FoxPro 数据库)类型数据源首选的访问接口;RDO(Remote Data Object,远程数据对象)是 VB 访问关系型 ODBC(Open Database Connectivity,开放的数据库连接)数据源(如 Microsoft SQL Server)的最佳访问接口。这两种方法都完整地集成于 VB 中,并且形成了 VB 核心的数据库访问策略。

ADO(Active Data Objects,活动数据对象)是 Microsoft 最新的独立于开发工具和开发语言的简单易用的数据接口,所有的数据源都可以通过 ADO 来访问。ADO 可用于各种程序设计语言,包括像 VBScript 和 Active Server Pages 这样的语言。因此,ADO 也是开发基于 Web 数据库应用的最佳技术。VB 6.0 的数据库应用程序体系结构如图 10-13 所示。

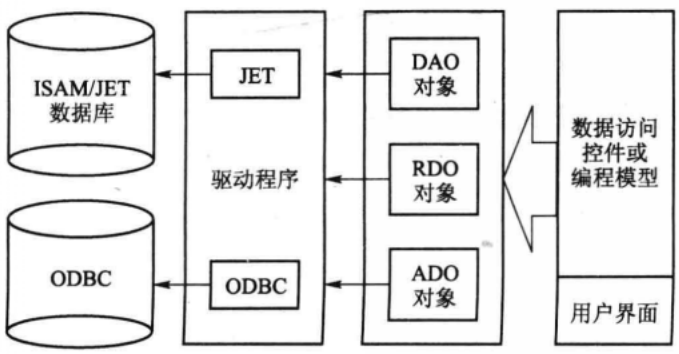


图 10-13 Visual Basic 6.0 的数据库应用程序体系结构

在数据库应用程序中,用户程序通过 DAO/RDO/ADO 对象(或封装该对象的数据控件,如 Data 控件)与被访问的数据库(数据源)建立连接,再通过编程直接访问 DAO/RDO/ADO 对象(或用数据访问控件,如 DBList 数据控件)来访问数据库中的数据。

10.4.2 ADO 控件

DAO 方式是允许程序员操纵 Microsoft Jet 数据库引擎的第一个面向对象的接口。对于单一系统的数据库应用程序来说,DAO 非常有效;但在中等以上规模的网络中,DAO 则不能胜任。ADO 方式是 DAO 和 RDO 方式的继承者,是 Microsoft 最新的对通用数据的访问接口,为所有数据源提供了高性能的访问,这些数据源包括关系和非关系型数据库、电子邮件和文件系统、文本和图形等,它更适合于大流量和大事务量的网络计算机系统。

要创建一个客户或前端数据库应用程序,应在窗体中添加 ADO 控件 Adodc 以及 ADO 数据

访问控件。添加 ADO 控件及 ADO 数据访问控件方法是在工具箱中上点鼠标右键，选择部件，然后选中“Microsoft ADO Data Control 6.0(OLE DB)”(添加 Adodc 控件)、“Microsoft DataList Control 6.0(OLE DB)”(添加 DataList、DataCombo 控件)、“Microsoft DataGrid Control 6.0(OLE DB)”(添加 DataGrid 控件)、“Microsoft Hierarchical FlexGrid Control 6.0(OLE DB)”(添加 MSHFlexGrid 控件)，这些控件如图 10-14 所示。

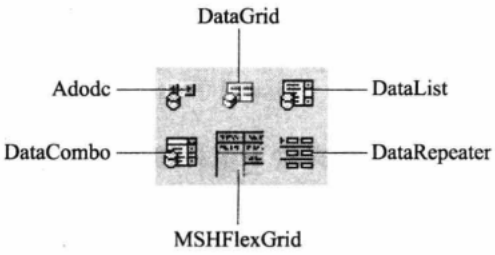


图 10-14　ADO 控件

ADO Data 控件 Adodc 与 Data 控件相似，它使用户能使用 ADO 快速地创建一个到数据库的连接，它的用法也与 Data 控件的用法相同。Adodc 控件的功能有：

① 连接一个本地数据库或远程数据库。

② 打开一个指定的数据库表，或定义一个基于结构化查询语言(SQL)的查询、存储过程，或获取该数据库中表的记录集合。

③ 将数据字段的数值传递给数据绑定的控件，可以在这些控件中显示、更改或添加新记录。

1. Adodc 控件的主要属性和方法

(1) ConnectionString 属性

指定有效的与数据源连接的字符串，通过该字符串使 Adodc 控件与指定的数据库建立连接，其格式见 10.5 节。它亦可通过对话方式设置，方法如下。

① 单击 Adodc 控件属性页的 ConnectionString，进入图 10-15 所示的“属性页”对话框。在该对话框中，用户可选择连接数据源，这里选择“使用连接字符串”，单击“生成”按钮，进入“数据链接属性”对话框的“提供者”选项卡(图 10-16)。

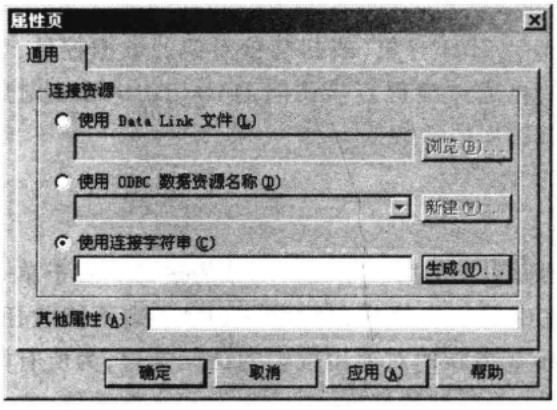


图 10-15　设置 Adodc 控件的 ConnectionString 属性



图 10-16　数据链接属性对话框

② 在“提供者”选项卡上，选择“Microsoft Jet 3.51 OLE DB Provider”，并单击“下一步”，进入

"连接"选项卡。

③ 在"连接"选项卡上选择或输入 Access 数据库名"D:\Xscjgl\Xscjgl.mdb",单击"测试连接"按钮可以测试设置是否正确。

④ 若正确,可单击"确定"按钮关闭对话框回到"属性页"对话框,生成的连接字符串出现在编辑框中,关闭此对话框。

(2) RecordSource 属性

设置或返回记录集的 SQL 查询语句或表名。它的设置方法如下:

单击属性页 Adodc 控件的 RecordSource,进入如图 10-17 所示的"属性页"对话框。在该对话框中,可设置记录源的命令类型(同 CommandType)、表或 SQL 语句文本。单击"确定"按钮关闭对话框。

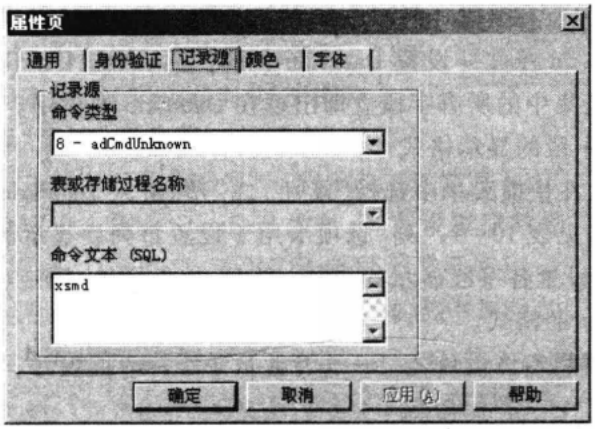


图 10-17 设置记录源对话框

(3) Refresh 方法

打开或重新打开连接的数据源,并能重建控件的 Recordset。

2. Adodc 控件的主要事件

(1) WillMove/MoveComplete 事件

更改 Recordset 中的当前记录指针位置前/后引发。

(2) WillChangeRecord/RecordChangeComplete 事件

在 Recordset 中的一个或多个记录更改前/后引发。

(3) WillChangeField/FieldChangeComplete 事件

对 Recordset 中的一个或多个 Field 对象值进行更改前/后引发。

10.4.3 ADO 数据访问控件

与 Data 控件一样,VB 内置的具有数据识别能力的控件都可作为 ADO 的数据访问控件。使用时,只需将 DataSource 设置为 Adodc 控件名即可。此外 VB 6.0 中新加入有 DataGrid、DataList、DataCombo、MSHFlexGrid 数据访问控件。其中 DataList 和 DataCombo 与 DBList 和 DBCombo 用法相似,下面仅介绍 DataGrid 控件的用法。

DataGrid 控件也称网格控件,它以电子表格的形式显示数据,允许对 Recordset 对象中代表记录和字段的一系列行和列进行数据操纵,它的主要属性如表 10-9 所示。

表 10-9　DataGrid 控件的主要属性

属 性 名	设置值及意义
DataSource	指定所连接的数据控件
AllowAddNew	设置是否允许增加新记录
AllowDelete	设置是否允许删除当前记录(用 Shift + Delete 键)
AllowUpdate	设置是否允许更新记录
Caption	设置网格顶端的标题文字

因 DataGrid 是针对整个记录集的,实际使用时,还可用下面的方法进一步设置,以满足在界面设计时的需要。

(1) 加入记录集中的所有字段

在窗体中建立 DataGrid 控件并设置 DataSource 属性后,右击 DataGrid 控件,在快捷菜单中选择"检索结构"项,记录集中的所有字段立即出现在 DataGrid 控件中。

(2) 设置记录集各字段的显示格式

右击 DataGrid 控件,在快捷菜单中选择"属性..."项,进入 DataGrid 属性设置对话框。"通用"选项卡中的项目可参见表 10-9;"列"选项卡用于设置每列要显示的字段及标题(顺序可更改);"布局"选项卡中可设置各字段显示的风格、宽度、是否可见、锁定(即只读)等。"格式"选项卡中可设置各字段显示的格式。

【例 10-3】 数据编辑窗体设计之二——专业目录维护界面设计。专业目录表 ZYML 的结构如表 10-3 所示。

设计过程如下:

① 新建一工程名为 VB10_3. VBP,建立一个窗体名为"frmZyml10_3. frm"窗体,加入相关控件,所需控件及其属性的设置如表 10-10 所示。该窗体运行时的界面如图 10-18 所示。

表 10-10　"专业目录维护界面"主要控件及其相关属性的设置

控 件 名	属 性 名	设 置 值	备　注
Adodc	Name	Adodc1	
	ConnectionString	"Provider = Microsoft. Jet. OLE DB. 3. 51; Data Source = D:\Xscjgl\xscjgl. mdb"	
	RecordSource	ZYML	
DataGrid	Caption	专业目录	
	DataSource	Adodc1	
	AllowAddNew	True	允许增加记录
	AllowDelete	True	允许删除记录
	AllowUpdate	True	允许修改记录

② 右击 DataGrid 控件,在弹出菜单中选择"检索字段",表中各字段名显示在网格中。

③ 右击 DataGrid 控件,在弹出菜单中选择"属性..."项,进入"属性"对话框。在"列"选项卡中设置每列(字段)显示的标题分别为"专业代码"、"专业名称"。在"布局"选项卡中,设置每列合适的宽度。

④ 运行该工程程序显示如图 10-18 所示。

【例 10-4】 完成案例十二如图 10-12 所示的学生档案管理窗体设计。



图 10-18 专业目录维护

在窗体中添加一个设置了两个选项卡的选项卡控件,“记录方式”选项卡上可对记录进行编辑维护,“浏览方式”选项卡上有一 DataGrid 控件以实现数据的浏览。程序运行时的界面如图 10-12 所示。

设计过程如下:

① 新建名为 VB10_4. VBP 工程文件,在工程中建立一名为 frmado10_4. frm 的窗体,在窗体中放置一选项卡控件 SSTab(若工具箱中没有,则通过执行〖工程〗→〖部件... 〗菜单命令将“Microsoft Tabbed Dialog Control 6.0”控件加入),右击该选项卡控件,选择“属性”菜单项,在“通用”选项卡上,设置“选项卡数”为 2,选项卡标题分别为“记录方式”和“浏览方式”。

② 在“浏览方式”选项卡上加一 DataGrid 控件,并在设置完数据控件 adoXsmd 后设置它的属性(数据源、列标题及布局、显示格式,置 BJDM 字段为不可见)。

③ 按图 10-12 所示加入其他所需控件,主要控件的关键属性的设置如表 10-11 所示。其中 adoXsmd 为 DBCombo 控件的数据源。

表 10-11 “学生档案管理”主要控件及其相关属性的设置

控件名	属性名	设置值
SSTab	Tabs	2
Adodc	Name	adoXsmd
	ConnectionString	Provider = Microsoft. Jet. OLE DB. 3. 51; Data Source = D:\Xscjgl\xscjgl. mdb
	RecordSource	Xsmd
Adodc	Name	Adodc2
	ConnectionString	Provider = Microsoft. Jet. OLE DB. 3. 51; Data Source = D:\Xscjgl\xscjgl. mdb
	RecordSource	BJMC
	Visible	False
DataGrid	DataSource	adoXsmd
	AllowDelete	False
	AllowUpdate	False
	AllowAddNew	False
CheckBox	DataSource	adoXsmd
	DataField	XB
DataCombo	DataSource	adoXsmd
	DataField	BJDM
	RowSource	Adodc2
	ListField	BJMC
	BoundColumn	BJDM

④ 各控件的事件代码如下：

```
Option Explicit
Dim SelectString As String
Private Sub SetButtons(bVal As Boolean)                    '设置按钮的状态
    cmdAdd. Enabled = bVal
    cmdDelete. Enabled = bVal
    cmdUpdate. Enabled = Not bVal
    cmdCancel. Enabled = Not bVal
End Sub
```

```
Private Sub cmdAdd_Click( )
  Dim strRxsj As String
  Dim strXh As String
  strXh = Text1. Text
  strRxsj = Text5. Text
  adoxsmd. Recordset. AddNew
  Text5. Text = strRxsj
  Text1. Text = strXh
  Text1. SetFocus
  SendKeys "{End}"
  SetButtons False
End Sub
```

```
Private Sub cmdCancel_Click( )
    adoxsmd. Recordset. CancelUpdate
    SetButtons True                                        '取消修改或增加记录
End Sub
```

```
Private Sub cmdClose_Click( )
   Unload Me
End Sub
```

```
Private Sub cmdDelete_Click( )
    adoxsmd. Recordset. Delete                             '删除当前记录
End Sub
```

```
Private Sub cmdexit_Click( )
Unload Me
End Sub
```

```
Private Sub cmdUpdate_Click( )
    adoxsmd. Recordset. UpdateBatch adAffectAll
    SetButtons True                                        '更新当前记录
End Sub
```

10.5 用编程方法引用 ADO 对象简介

与 Data 控件一样,使用 Adodc 控件不能完成复杂的数据库应用系统设计的需要(如使用 SQL 的 Delete 语句),而常常要对 ADO 对象进行编程。ADO 对象模型的结构如图 10-19 所示。它包含以下几个主要对象。

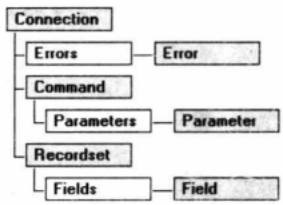


图 10-19 ADO 对象模型

1. ADO 的 Connection 对象

Connection 对象是一个主要的顶层 ADO 组件,必须成功地打开一个连接到数据源的 Connection 对象,才有可能使用相关的 Command 对象或 Recordset 对象。

Connection 对象可以指定数据源,建立与数据源的连接,执行 SQL 语句。如果是客户端/服务器数据库系统,该对象可以等价于到服务器的实际网络连接。

(1) Connection 对象主要属性

ConnectionString 属性:用于设置建立连接的数据源的信息(字符串)。它是用由一系列分号分隔的"= value"语句的字符串来指定数据源。ADO 支持的 ConnectionString 属性参数如表 10-12所示。如在图 10-15 中通过对话框取得的 ConnectionString 属性值为:"Provider = Microsoft. Jet. OLEDB. 3. 51;Data Source = D:\Xscjgl\xscjgl. mdb"。

表 10-12 ConnectionString 属性参数

参数(argument)	说 明
Provider	指定用来连接的提供者名称
Data Source	指定包含预先设置连接信息的特定提供者的文件名称
Remote Provider	指定打开客户端连接时使用的提供者名称(仅限于 Remote Data Service)
Remote Server	指定打开客户端连接时使用的服务器的路径名称(仅限于 Remote Data Service)

(2) Connection 对象的主要方法

① Open 方法用于建立与数据源的物理连接,连接完成后可对它发出命令且处理结果。用法如下:

```
connection. Open ConnectionString, User ID, Password, OpenOptions
```

其中各参数的含义如下。

- ConnectionString:可选,字符串,包含连接信息。
- User ID:可选,字符串,包含建立连接时所使用的用户名称。
- Password:可选,字符串,包含建立连接时所用密码。
- OpenOptions:可选,设置连接方式为异步还是同步,默认为同步连接。

② Execute 方法用于执行指定的查询、SQL 语句等内容。用法如下:

- 对于没有返回记录集的命令(如更新、插入、删除记录的 SQL 命令)字符串:

```
connection. Execute  CommandText, RecordsAffected, Options
```

- 对于具有返回记录集的命令(如 SQL 查询)字符串:

```
Set recordset = connection. Execute (CommandText, RecordsAffected, Options)
```

其中各参数的含义如下。

- CommandText:字符串,包含要执行的 SQL 语句、表名、存储过程等的文本。
- RecordsAffected:可选,长整型变量,提供者向其返回操作所影响的记录数目。
- Options:可选,指示应如何为 CommandText 参数赋值,如表 10-13 所示。

表 10-13　Options 参数可选的常量

常　量	说　明
AdCmdText	指示应将 CommandText 赋值为命令的文本定义,如 SQL 语句
AdCmdTable	指示 ADO 应生成 SQL 查询以便从 CommandText 命名的表中返回所有行
AdCmdTableDirect	指示应从 CommandText 命名的表中返回所有行
AdCmdStoredProc	指示应将 CommandText 赋值为存储过程
AdCmdUnknown	指示 CommandText 中的命令类型未知(默认)
AdExecuteAsync	指示命令应该异步执行
AdFetchAsync	指示 CacheSize 属性指定的初始数量之后的行应异步提取

③ Close 方法用于关闭打开的对象及任何相关对象。

(3) 编程模式下 ADO 对象的使用方法

① 在工程中添加对"Microsoft ActiveX Data Objects 2.0 Library"的引用,以便在程序中引用 ADO 对象。

② 声明 ADO 对象变量。

③ 创建 ADO 对象的引用。

④ 使用 ADO 对象变量。

⑤ 结束时释放 ADO 对象变量。

典型的代码如下:

```
Dim  strCnn  As  String
Dim  cnn  As  ADODB. Connection      '声明 Connection 对象变量
strCnn = "Provider = Microsoft. Jet. OLEDB. 3. 51;" & "Data Source D:\Xscjgl\xscjgl. mdb"
                                     '注意 Data 与 Source 间仅有一个空格
Set cnn = New  ADODB. Connection     '创建 Connection 对象的引用
cnn. Open  strCnn                    '打开连接
...                                  '使用对象变量 cnn 的代码
cnn. Close                           '关闭连接
Set  cnn = Nothing                   '释放资源
```

2. ADO 的 Command 对象

Command 对象定义了将对数据源执行指定的命令。使用 Command 对象查询数据库并返回 Recordset 对象中的记录,或处理数据库结构。

(1) Command 对象的主要属性

- ActiveConnection 属性:设置或返回指定的 Command 对象当前所属的 Connection 对象。

- CommandText 属性:设置或返回命令(如 SQL 语句、表名)的字符串值。
- CommandType 属性:指示 Command 对象的类型。它的常用取值如表 10-13 中前 5 个。

如果 CommandType 与 CommandText 属性中的命令类型不匹配,调用下面的 Execute 方法执行命令时将产生错误。

(2) Command 对象的主要方法

Execute 方法用于执行在 CommandText 中指定的查询、SQL 语句。其用法如下:

- 对于没有返回记录集的命令(如更新、插入、删除记录的 SQL 命令)字符串:

```
command. Execute  RecordsAffected, Parameters, Options
```

- 对于具有返回记录集的命令(如 SQL 查询)字符串:

```
Set recordset = command. Execute( RecordsAffected, Parameters, Options )
```

其中各参数的含义如下。

- RecordsAffected:可选,长整型变量,提供者向其返回操作所影响的记录数目。
- Parameters:可选,变体型数组,使用 SQL 语句传送的参数值。
- Options:可选,长整型值,指示提供者如何对 Command 对象的 CommandType 属性赋值,见 Connection 的 Execute 方法。

☞ 对于具有返回记录集的 Execute 方法,也可通过将查询字符串传送给 Connection 对象的 Execute 方法或 Recordset 对象的 Open 方法来实现。

3. Recordset 对象

10.3.3 节介绍了 DAO Recordset 对象的部分属性与方法,ADO 模型中的 Recordset 对象还有几个常用的属性和方法。

① Source 属性用于指示 Recordset 对象中数据的来源(可以是 Command 对象、SQL 语句或表的名称)。

② ActiveConnection 属性用于设置或返回指定的 Recordset 对象当前所属的 Connection 对象。

③ Open 方法用于打开表、查询结果或者以前保存的 Recordset 中记录集。其用法为:

```
recordset. Open  Source, ActiveConnection, CursorType, LockType, Options
```

其中各参数的含义如下。

- Source:可选。变体型,可以是 Command 对象变量名、SQL 语句或表名。
- ActiveConnection:可选。变体型,可以是 Connection 对象变量名或含 ConnectionString 参数的字符串。
- CursorType:可选。确定打开 Recordset 时应该使用的光标类型。
- LockType:可选。确定打开 Recordset 时应使用的锁定(并发)类型值。
- Options:可选。长整型值,指示如何计算 Source 参数,见 CommandType 属性。

④ UpdateBatch 方法用于更新记录集。用法为:

```
recordset. UpdateBatch  AffectRecords
```

其中,参数 AffectRecords 是可选的,它决定 UpdateBatch 方法所影响的记录数目,通过选取 adAffectAll(默认值)对 Recordset 对象的所有记录进行更新。

编译时如果提示“用户定义类型未定义”错误，可在窗体上放置一个 Adodc 控件，然后删除它即可，或通过 VB 菜单的〖工程〗→〖引用...〗命令打开对话框，在工程中添加对“Microsoft ActiveX Data Objects 2.0 Library”的引用。

在应用程序中直接利用 ADO 对象进行编程的方法为：

方法一：使用 Connection 对象。

① 利用 Connection 对象的 ConnectionString 属性设置与数据库进行连接的信息串，用它的 Open 方法与数据库建立物理连接。

② 如果要执行 INSERT、UPDATE、DELETE 等不返回记录集的 SQL 命令，使用不返回记录集的 Execute 方法，SQL 命令串作为 Execute 方法的参数。

③ 如果要执行 SELECT 查询返回记录集的 SQL 命令，使用返回记录集的 Execute 方法，SQL 命令串作为 Execute 方法的参数，并将返回的记录集传送给 Recordset 对象。

方法二：使用 Command 对象。

① 利用 Connection 对象的 ConnectionString 属性设置与数据库进行连接的信息串，利用它的 Open 方法与数据库建立物理连接。

② 设置 Command 对象的 ActiveConnection 属性为①中建立的 Connection 对象。

③ 设置 Command 对象的 CommandText 属性及 CommandType 属性。

④ 如果要执行 INSERT、UPDATE、DELETE 等不返回记录集的 SQL 命令，使用不返回记录集的 Command 对象的 Execute 方法。

⑤ 如果要执行 SELECT 查询返回记录集的 SQL 命令，使用返回记录集的 Command 对象的 Execute 方法，并将返回的记录集传送给 Recordset 对象。

【例 10-5】 用编程的方法引用 ADO 对象，设计出在库表 ZYML 中根据专业代码查找专业名称，并在右边的两个文本框中分别显示专业代码和专业名称的应用程序窗体（图 10-20）。

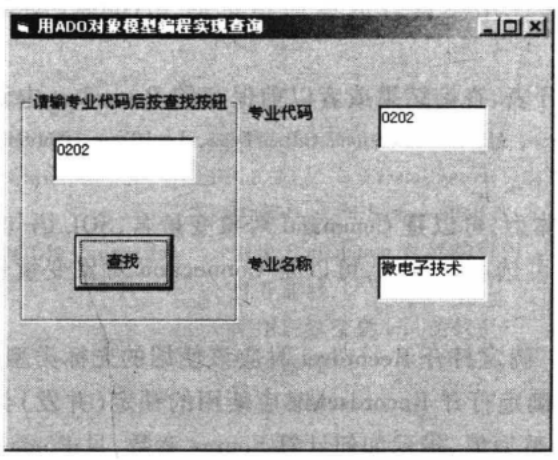


图 10-20　用编程的方法引用 ADO 对象

设计过程如下：

① 新建名为 VB10_5. VBP 工程文件，在工程中建立一名为“frmadobc10_5. frm”的窗体。

② 通过操作“工程”菜单下的“引用”菜单的对话框，在工程中引用“Microsoft ActiveX Data

Objects 2.0 Library",以便编程引用 ADO 对象。

③ 按图 10-20 所示加入所需控件,主要控件的关键属性的设置如表 10-14 所示。

表 10-14 "用编程的方法引用 ADO 对象示例"主要控件及其相关属性的设置

控 件 名	属 性 名	设 置 值
框架	Name	Frame1
	Caption	请输专业代码后按查找按钮
框架内的命令按钮	Name	cmdfind
	Caption	查找
框架内的文本框	Name	Txtfind
	Text	空
专业代码文本框	Name	txt_dm
	Text	空
专业名称文本框	Name	txt_mc
	Text	空

④ 查找按钮的 Click 事件代码如下:

```
Private Sub cmdfind_Click( )            '按学号查找记录
  Dim cnn2 As ADODB. Connection
  Dim rs2 As ADODB. Recordset
  Dim s As String
  Set cnn2 = New ADODB. Connection
  cnn2. Provider = "microsoft. jet. oledb. 3. 51"
  cnn2. Open "D:\xscjgl\xscjgl. mdb"
  Set rs2 = New ADODB. Recordset
  s = "select * from zyml where zydm = '" & txtfind. Text & "'"
  rs2. Open s, cnn2, adOpenDynamic
    If Not rs2. EOF Then
      txt_dm = rs2! zydm
      txt_mc = rs2! zymc
    End If
    If rs2. EOF Then
       txt_dm = ""
       txt_mc = ""
       MsgBox "没找到!"
    End If
  End Sub
```

习 题 10

1. 什么是数据库、数据库管理系统、数据库应用程序、数据库系统?

2. 什么是关系型数据库? 什么是表? 表是由哪些成分组成的? 数据表的字段含有哪些信

息？什么是主关键字？常见的数据类型有哪些？

3. 使用 VisData 或 Microsoft Access 建立一个数据库 Xscjgl. mdb，将它存放在 D:\Xscjgl 文件夹下，并按本章的表结构要求建立所有表的结构及相应的索引。

4. Data/Adodc 控件的主要属性有哪些？主要方法有哪些？Adodc 的主要事件有哪些？分别在何种情况下使用？试说明用 Data/Adodc 控件编写数据库应用程序的步骤。

5. ADO 对象模型中 Connection、Command、Recordset 对象的主要属性、方法有哪些？简述编程方式下使用这些对象的方法。

6. 总结使用 ADO 对象模型从 Xscjgl. mdb 中获得表 User 对应记录集的方法（提示：可用 Connection 和 Command 对象的 Execute 方法，Recordset 对象的 Open 方法）。

7. 利用 Data 控件建立用于维护 KCMC 表的窗体。

8. 利用 Adodc 控件建立用于维护 KCMC 表的课程编码维护窗体，窗体命名为“frmKcbm”。

9. 利用 Adodc 控件建立用于维护 USER 表的用户管理窗体，窗体命名为“frmUser”。

10. 试编写一个电话号码查询程序，要求能够根据用户的区域、公司类别等特性，进行模糊查询（使用 Like 运算符）。

11. 上机完成本章中的 5 个例题。

附录 实　验

实验的总体要求与方法

实验是巩固课堂所学知识、掌握可视化程序设计方法、提高分析问题和解决问题能力的最重要的途径。因此明确实验目的、要求、实验步骤及实验规则，一方面有助于实验任务的顺利完成，另一方面也有利于养成严谨科学的作风。每个实验都要遵循认真预习准备、认真做好实验、认真写好实验报告三个阶段进行。

1. 实验的基本目的

① 通过实验熟悉面向对象程序设计的基本概念。

② 通过实验熟练掌握利用 VB 进行可视化程序设计的基本方法。

③ 通过实验掌握应用程序的调试方法，加强应用能力的训练与培养，从而具有设计一般应用程序和解决事务管理问题的能力。

2. 实验的基本要求

① 上机实验之前，应认真预习有关实验内容及相应教材，写出实验步骤（对程序设计实验题还应画出流程图、编制好源程序、准备原始数据及写出调试步骤）。

② 实验过程中应记录实验中的有关数据（包括出错信息等），以便课后进行分析。

③ 所有实验都应具备下列条件：

- 硬件：PC 及其兼容机即可。
- 软件：操作系统为中文 Windows 系列，开发环境为 Visual Basic 6.0 中文企业版及 Microsoft Developer Network（MSDN）。

④ 除非特别指明，约定每个实验的相关窗体文件或模块文件都存放在“D:\Vbsy”文件夹下，窗体文件的命名规则为“frmSy_xx_xx. frm”，模块文件的命名规则为“Sy_xx_xx. bas”，如实验题 6－1 的窗体文件命名为“frmSy_6_1. frm”。

⑤ 上机时要遵守实验室的规章制度，爱护实验设备。

⑥ 本附录中，每个实验都有难度不等的若干个实验题，读者可根据实际情况选做其中的一部分，或将每章后面的习题作为实验题。

3. 实验报告

每个实验结束后都应写出实验报告，报告中应包括下列内容：

① 实验目的。

② 实验内容与要求。

③ 实验步骤（调试说明、上机遇到的问题及解决方法，观察到的现象及其分析）。

④ 源程序与流程图。

⑤ 实验结果及分析讨论，实验心得体会等。

实验 1 VB 环境的使用和简单应用程序设计

一、实验目的

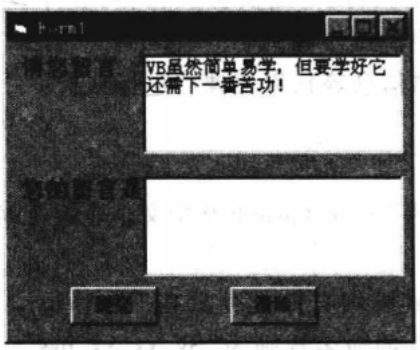
1. 掌握应用软件的安装方法。
2. 熟悉 Visual Basic 6.0 开发环境。
3. 掌握建立、编辑和运行一个简单 VB 应用程序的方法。
4. 掌握常用控件(标签、文本框、命令按钮)的应用。
5. 掌握使用 MSDN 获取帮助的方法以及通过 Internet 获取帮助的方法。

二、实验内容与要求

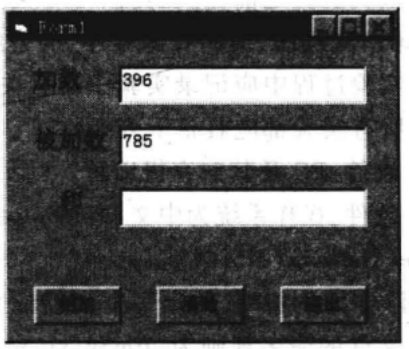
1. 安装 Visual Basic 6.0 中文版及 Microsoft Developer Network(MSDN)。
2. 利用 Visual Basic 6.0 集成开发环境编写简单的 VB 应用程序。

① 按附图 -1 所示界面设计一个留言簿。要求单击“提交”按钮后,上面的文本框内容复制到下面的文本框中,且下面文本框中的内容不能修改(只读)。单击“清除”按钮则清除两个文本框中的内容。

② 按附图 -2 所示界面设计一个加法器。



附图 -1 留言簿



附图 -2 简易加法器

3. 通过 MSDN 获取关于 TextBox 控件的属性、方法和事件的信息。
4. 通过 IE 浏览器,访问以下 Internet 站点:

① Visual Basic 开发中心(http://msdn.microsoft.com/vbasic/)。

② 全球最大中文 IT 社区(http://www.csdn.net/)。

实验 2 顺序结构与选择结构

一、实验目的

1. 掌握 VB 中标准数据类型的声明、使用方法。

2. 掌握常用函数和 InputBox 与 MsgBox 的使用。

3. 会使用判定结构和选择结构编制应用程序。

二、实验内容与要求

下列程序设计题可在窗体的 Click 事件中编写代码，并用 Print 方法将结果输出到窗体上。

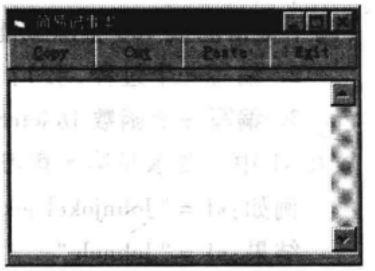
1. 设计一个程序，根据输入数组元素的下标，输出对应元素的值。要求程序能判断下标值是否正确，若越界则给出错误信息。

2. 设计一个程序，输入某人的工资，计算应缴纳的个人所得税。应缴个人所得税的计算方法为：每月应纳税额减去 3 500 元后的余额为当月计税额，按下列标准赋税：

应纳税 = X × 税率 - 速算扣除数

级　数	全月应纳税所得额 x/元	税率/%	速算扣除数/元
1	不超过 1 500	3	0
2	1 500 ~ 4 500	10	105
3	4 500 ~ 9 000	20	555
4	9 000 ~ 35 000	25	1 005

3. 按附图 -3 设计一个简易记事本。要求命令按钮 Copy、Cut、Paste 和 Exit 能实现文本信息的“复制”、“剪切”、“粘贴”和“退出”功能。



附图 -3　简易记事本

三、分析与讨论

1. 比较单行结构与块结构的条件语句的特点。

2. 比较 IF 条件语句与 Select 条件语句的差别。

实验3　循环结构

一、实验目的

1. 掌握 3 种基本循环语句的使用及应用场合。

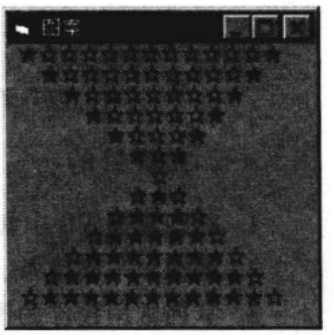
2. 熟悉各种算法，能使用循环语句设计应用程序。

二、实验内容与要求

1. 试用第 4 章中的各种循环实现求：$S = 1 + 3 + \cdots + 99$。

2. 试编写一程序实现附图 -4 所示的图案。

3. 编一程序显示所有水仙花数。所谓水仙花数，是指一个 3 位数，其各位数字立方和等于该数字本身。例如，153 是水仙花数，因为 $153 = 1^3 + 5^3 + 3^3$。



附图 -4　输出图案

4. 编程打印九九乘法表。

三、分析与讨论

1. 如何控制循环条件,以防止死循环或不循环?
2. 试总结各种循环语句的适用场合。

实验 4 过程的使用

一、实验目的

1. 掌握函数与过程的定义和调用方法。
2. 理解变量、函数和过程的作用域。
3. 掌握参数按值传递和按地址传递的方法。

二、实验内容与要求

1. 编写一个 Function 过程实现判断某一年是否为闰年,如"是"返回 True,否则返回 False。(闰年是指能被 400 整除或能被 4 整除但不能被 100 整除的年份)

2. 编写一个过程,求 1! +2! +… +10!。

3. 编写一个函数 DeleteStr(s1,s2),实现将字符串 s1 中出现的子字符串 s2 删去,结果仍存放在 s1 中。要求见第 5 章习题 5.10。

例如:s1 = "Johnjokel joke",s2 = "jok"

结果:s1 = "Johnele"

三、分析与讨论

1. Sub 过程与 Fuction 过程有何区别?

2. 在"通用"部分定义的变量作用范围是什么? 在 Sub 过程或 Fuction 过程内定义的变量作用范围是什么?

3. 过程的参数传递采用值传递与地址传递有何区别?

实验 5 常用控件应用之一

一、实验目的

1. 熟练掌握 VB 中常用控件(单选框与检查框、列表框与组合框等)的使用方法。
2. 掌握水平滚动条、垂直滚动条和定时器控件的使用方法。
3. 掌握常用控件的事件功能和事件代码的编写方法。
4. 进一步理解和掌握事件编程的方法,掌握利用事件代码控制应用程序运行的方法。

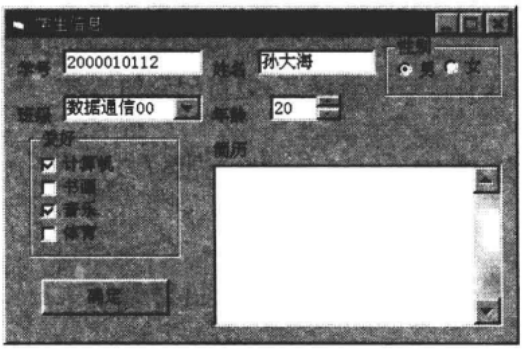
二、实验内容与要求

1. 完善实验 2 中简易记事本的程序设计，要求利用剪贴板来完成指定的功能。

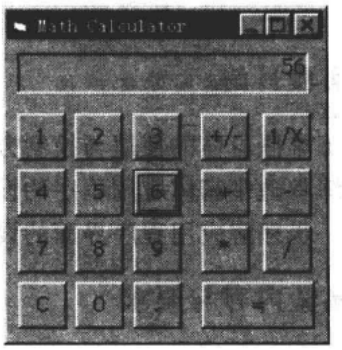
【提示】 利用 VB 中的剪贴板对象 ClipBoard 的 GetText 方法，返回 ClipBoard 对象中的文本字符串。粘贴可用 txtNote. Text = Left(txtNote. Text, txtNote. SelStart) + ClipBoard. GetText + Right(txtNote. Text, Len(txtNote. Text) - txtNote. SelStart) 来实现。

2. 设计一个用于输入学生信息的应用程序，运行时的界面如附图 -5 所示。

*3. 按附图 -6 所示界面设计一个简易计算器（图中的 C 键为清除键）。



附图 -5　学生信息录入窗体



附图 -6　简易计算器

【提示】

① 将数字键定义为控件数组，设置 4 个全局变量 Operand1、Operand2、Operator、ClearDisplay 分别用来存放 2 个操作数、1 个运算符和 1 个当前运算结束标志（布尔型）。

② 单击数字键时，保存输入的数字。

```
lblDisplay. Caption = lblDisplay. Caption + cmd Digits( Index). Caption
```

③ 单击“ = ”键时，根据运算符进行运算。

```
Private Sub cmdEquals_Click( )
    Dim result As Double
    Operand2 = Val( lblDisplay. Caption)
    If Operator = " + " Then result = Operand1 + Operand2
    If Operator = " - " Then result = Operand1 - Operand2
    If Operator = " * " Then result = Operand1 * Operand2
    If Operator = "/" And Operand2 < > "0" Then result = Operand1 / Operand2
    lblDisplay. Caption = result
    ClearDisplay = True
    'ClearDisplay 为全局变量，本次运算结束，设置清除标志，以便输入新的操作数。
End Sub
```

④ 单击运算符键时，保存输入的第一操作数及运算符，清除显示的数据。

```
Private Sub cmdPlus_Click( )
    Operand1 = Val( lblDisplay. Caption)
    Operator = " + "
    lblDisplay. Caption = " "    '清除显示的数据
End Sub
```

三、分析与讨论

1. ComboBox 控件的 Click、Change 事件在何时发生？有何区别？

2. 如何使控件在运行时不可见？或变灰不可使用？

3. 将本实验中对学生信息录入窗体的要求改为在输入合法的学号后(规定 6 位数字字符),窗体上的其他控件才可使用,应利用什么事件实现？怎样修改？

实验 6 常用控件应用之二

一、实验目的

1. 熟练掌握窗体的使用与控制方法。

2. 熟练掌握多窗体应用程序的设计方法。

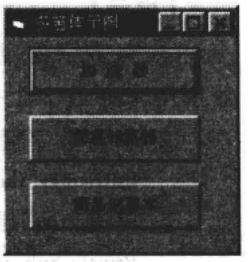
3. 进一步理解应用程序界面的设计原则与方法。

二、实验内容与要求

1. 对实验 5 中的简易记事本程序进行修改,要求先进行注册,是合法用户方可使用。

【提示】 注册窗体生成可通过选择〖工程〗→〖添加窗体〗命令菜单,打开“添加窗体”对话框,并在其中选择“登录对话框”由 VB 自动实现,生成的窗体名为“frmLogin. frm”,将它存放在“D:\Xscjgl”文件夹中。

2. 按附图 -7 界面设计一个窗体,以便通过单击上面的命令按钮分别调用实验 1、实验 2、实验 5 的窗体。要求运行时按模式方式显示被调用的窗体。



附图 -7 多窗体界面

三、分析与讨论

1. 窗体的 Action 事件和 Load 事件分别在何时发生？利用这两个事件可完成什么任务？

2. 模式窗体和非模式窗体有什么区别？

实验 7 键盘与鼠标

一、实验目的

1. 熟悉与掌握键盘事件、鼠标事件的触发时机、使用方法。

2. 学会在不同的应用要求中选择不同的事件。

3. 理解拖放的概念、拖放的编程要点,会进行简单拖放的程序编制。

二、实验内容与要求

1. 设计一个程序,在一个多行文本框中输出各个按键的 KeyCode 码和相应的 KeyAscii 码

(如存在)。

2. 编写程序,在窗体上画圆。要求:按着鼠标右键移动可画圆,否则不行。VB 中画圆的命令为"Circle (X,Y),R",表示以 X、Y 为圆心,以 R 为半径画一个圆。

3. 在窗体上放置两个图片框控件 picture1 和 picture2,在 picture1 中装入一幅图片。程序运行时拖动 picture1(拖动图标就是图片本身)刚进入、经过、离开 picture2 时,分别为 picture2 设置 3 种不同的前景色。

三、分析与讨论

1. 通过该实验,你能分析 KeyCode 码与 KeyAscii 码的区别与联系吗?借此说明 KeyPress 事件与 KeyDown 事件应用场合的不同。

2. 要实现鼠标拖放,其编程要点是什么?自动拖放与手工拖放有何区别?

实验8 菜 单

一、实验目的

1. 熟练掌握窗口菜单的规划、设计和使用方法。
2. 掌握弹出式菜单设计和使用方法。
3. 了解多文档 MDI 窗体的应用方法。

二、实验内容与要求

1. 根据第 7 章中关于"学生成绩管理系统"菜单的规划,设计该菜单系统。要求新建一名为 Xscjgl 的工程,存放在"D:\Xscjgl"文件夹下,工程的主窗体为含有菜单系统的窗体 MainForm。

2. 在实验 6 的基础上设计一弹出式菜单,要求用鼠标右击窗体时能选择调用"加法器"、"简易计算器"、"简易记事本"。

*3. 按 Windows 附件的"写字板"的菜单系统设计一个简易书写器 MDI 窗体。要求子窗体中使用 TextBox 作为文本文件的编辑器。

三、分析与讨论

1. 菜单系统的设计分几个步骤?
2. 单文档窗体 SDI 与多文档窗体 MDI 窗体各有什么特点?

实验9 工 具 栏

一、实验目的

1. 掌握工具栏、图像列表框控件的使用方法。

2. 掌握状态栏控件的使用方法。

3. 综合运用所学的可视化编程知识,编制具有可视化界面的应用程序。

二、实验内容与要求

1. 为实验 8 的“学生成绩管理系统”添加工具栏,工具栏上有“注册”、“注销”、“打印”3 个快捷按钮。并为它添加两个 Panel 状态栏,一个显示当前的工作状态,另一个显示系统键盘状态。

*2. 为实验 8 的简易书写器添加工具栏和状态栏,要求工具栏上含有与写字板相同的按钮,状态栏上有相同的 Panel。

实验 10 文 件

一、实验目的

1. 掌握文件管理控件的使用。

2. 掌握顺序文件和随机文件的打开、关闭和读写操作。

3. 掌握常用对话框的使用。

二、实验内容与要求

1. 按一般打开文件对话框的式样,用文件管理控件实现打开文件对话框的功能。

2. 有 10 个学生,每个学生有 3 门课程的成绩,从键盘上输入以上数据(包括学号,姓名,3 门课成绩),计算出平均成绩,将原有数据和计算出的平均分数存放在磁盘文件“stud”中。

3. 将上题“stud”文件中的学生数据,按平均分进行排序处理,将已排序的学生数据存入一个新文件“stu_sort”中。

4. 设计一个文本文件阅读器,其界面如附图 -8 所示。



附图 -8 文本文件阅读器

实验 11 图形处理与多媒体应用

一、实验目的

1. 掌握多媒体应用程序的设计方法。

2. 熟悉简单图形设计的方法。

3. 掌握简单动画设计的方法。

4. 初步掌握 ActiveX 控件的使用。

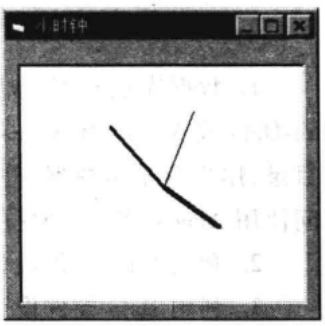
5. 熟悉动态网页设计的方法。

二、实验内容与要求

1. 设计一个类似 Windows 中的媒体播放器,以便能播放多种媒体文件。

【提示】 在工程中设计含有可播放不同媒体文件的多个窗体，根据打开的多媒体文件类型调用相应的窗体来播放该文件。

2. 设计一个时钟演示程序，运行时的界面如附图 -9 所示。



附图 -9　小时钟运行界面

【提示】

① 在窗体的 Load 事件中设定 3 种指针的长度：

```
Private Sub Form_Load( )
  nWidth = Picture1. Width -20
  nHeight = Picture1. Height -10
  nHourLen = nWidth * 4 / 18
  nMinLen = nWidth * 5 / 18
  nSecLen = nWidth * 6 / 18
End Sub
```

② 在定时器的 Timer 事件中编写代码，使时钟运动起来：

```
Private Sub Timer1_Timer( )
  Dim nSecond, nMinute, nHour As Integer
  Picture1. Cls                                    '清除图像
  nSecond = Second(Time( ))                        '取时、分、秒值
  nMinute = Minute(Time( )) + nSecond / 60
  nHour = (Hour(Time( )) Mod 12) + nMinute / 60 + nSecond / 3600
  Picture1. DrawWidth = 3                          '绘制时针，函数 Carc( )用于将角度转换为弧度
  Picture1. Line (nWidth / 2, nHeight / 2) - (nWidth / 2 - nHourLen * Sin(Carc _
    ((12 - nHour) * 30)), nHeight / 2 - nMinLen * Cos(Carc((12 - nHour) * 30)))
  Picture1. DrawWidth = 2                          '绘制分针
  Picture1. Line (nWidth / 2, nHeight / 2) - (nWidth / 2 - nMinLen * Sin(Carc _
    ((60 - nMinute) * 6)), nHeight / 2 - nMinLen * Cos(Carc((60 - nMinute) * 6)))
  Picture1. DrawWidth = 1                          '绘制秒针
  Picture1. Line (nWidth / 2, nHeight / 2) - (nWidth / 2 - nSecLen * Sin(Carc _
    ((60 - nSecond) * 6)), nHeight / 2 - nMinLen * Cos(Carc((60 - nSecond) * 6)))
End Sub
```

3. 完成第 9 章中的习题 5，使帆船在窗体中运动。

三、分析与讨论

1. 利用 VB 怎样进行简单图形设计？
2. 总结利用 ASP 访问数据库并将结果返回到客户端的设计步骤。

*实验 12　数据库的创建与数据控件

一、实验目的

1. 掌握利用 VisData 或 Microsoft Access 创建和维护 Access 数据库的方法。
2. 掌握利用 DAO/ADO 对象访问数据库的方法。

3. 会运用数据控件设计数据库访问界面。

二、实验内容与要求

1. 数据库的创建和维护。根据 10.1.4 节中学生成绩管理系统中的各个表结构,利用 VisData 或 Microsoft Access 创建数据库 Xscjgl,建立全部的数据表结构,并输入班级、学生、专业目录、用户等基本数据(注意,由于 VB 6.0 中仅能访问 Access 97 数据库,如果要在 VB 6.0 中访问使用 Access 2000 创建的数据库,必须使用 ADO/DAO 方式)。

2. 利用 Data 控件设计课程编码维护窗体,用于维护表 KCMC。

3. 利用 Data 控件设计学生成绩管理系统的注册窗体(frmLogin. frm),要求用户身份的验证通过表 USER 完成。在 MainForm 中调用本窗体,若是管理员,将具有编辑、系统维护菜单的操作权限,否则仅有查询菜单的操作权限。

【提示】 在注册窗体中,利用 Data 控件的 Recordset 属性的 Find 方法实现在表中查找指定的用户,若找到则取出 Password 字段数据,验证口令。若是合法用户,再取 Right 字段,判断是否为管理员。若是,则将编辑和系统维护菜单的 Enabled 置为 True,否则为 False。如 MainForm. menuSystem. Enabled = True。

4. 修改步骤 2,利用 Adodc 控件访问表 KCMC,窗体命名为“frmKcbm”,存放在“D:\Xscjgl”文件夹中。

三、分析与讨论

总结用 Data 控件和 Adodc 控件编写数据库应用程序的步骤。

参考文献

[1] Petroutsos E. Visual Basic 6.0 从入门到精通[M]. 北京:电子工业出版社,1999.

[2] Microsoft. Microsoft Visual Basic 6.0 控件参考手册[M]. 北京:希望电脑公司,1999.

[3] 龚沛曾,等. Visual Basic 程序设计教程[M]. 北京:高等教育出版社,1998.

[4] 廖彬山,等. Visual Basic 面向对象程序设计[M]. 北京:清华大学出版社,2000.

[5] 张海藩. 软件工程导论[M]. 北京:清华大学出版社,1987.

[6] 教育部考试中心. 全国计算机等级考试二级教程——Visual Basic 语言程序设计[M]. 北京:高等教育出版社,2001.